



Francisco
Martins

Sistema Telemantenção Industrial





**Francisco
Martins**

Sistema Telemantenção Industrial

Dissertação de Mestrado Integrado do curso de Engenharia Electrónica e de Telecomunicações, realizado sob a orientação científica do Professor Doutor António Pereira de Melo (Professor Catedrático), docente do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro. Contou ainda com a colaboração da Selmatron, através do Sr. Carlos Mendes.

o júri / the jury

presidente / president

Professor Doutor Alexandre Manuel Moutela Nunes da Mota
Professor Associado da Universidade de Aveiro

vogais / examiners committee

Professor Doutor António Ferreira Pereira de Melo
Professor Catedrático da Universidade de Aveiro (orientador)

Professor Doutor Carlos Alberto Caridade Monteiro Couto
Professor Catedrático da Universidade do Minho

Agradecimentos

É com muito gosto que aproveito esta oportunidade para agradecer ao Professor Doutor António Pereira de Melo, e ao Sr. Carlos Mendes por toda a disponibilidade, paciência e dedicação, ao longo de todo o desenvolvimento do projecto.

Um agradecimento muito especial a minha família por todo o apoio ao longo de todos estes anos.

Resumo

Este projecto visa a criação de um sistema de telemanutenção industrial, com a possibilidade de envio de notificações de alarmes que ocorram numa fábrica.

Através de uma rede industrial é possível adquirir os alarmes das diversas unidades de produção, essas anomalias são enviadas através de uma plataforma GSM, para o departamento de manutenção a fim de poder solucionar o problema.

É uma aplicação bastante importante pois procura minimizar os prejuízos de uma entidade fabril, causados por pequenas anomalias.

Este trabalho foi realizado no âmbito de uma parceria entre o DETI-UA e a Selmatron.

Palavras chave

Alarmes, Telemanutenção, LabVIEW, Selmatron

Abstract

This project aims build a system of industrial telemanutention, having the possibility to send alarm notifications that occurs on a factory.

Through a industrial network it's possible get alarms from sundry production units, these anomalies are sent via a GSM system, to the maintenance department, in order to they can solve the problem.

This is a very important application as it seeks to minimize the damage of a manufacturing entity, caused by small anomalies.

This work was conduced under a partnership between DETI-UA and Selmatron

Keywords

Alarms, Telemanutention, LabVIEW, Selmatron

Conteúdo

1	Introdução	1
1.1	Identificação dos objectivos a atingir	2
1.2	Enquadramento global	2
1.2.1	Alarmes	3
1.3	Enquadramento Temporal	5
1.4	Objectivo base do projecto	5
1.5	Importância do projecto	6
1.6	Organização do projecto	6
1.7	Condições iniciais de Desenvolvimento	7
1.7.1	O LabVIEW	7
1.7.2	O <i>hardware</i> , e <i>software</i> de apoio	8
2	Bases de Trabalho	9
2.1	Técnicas de Comunicação	10
2.1.1	Via de Comunicação	10
2.1.2	Comunicação com os operários	10
2.2	<i>State of the art</i>	10
2.2.1	Interfaces M2M	11
3	Componentes do Sistema	13
3.1	O LabVIEW	14
3.1.1	Programação por fluxo de dados	14
3.1.2	Programação Gráfica	15
3.1.3	Blocos, o conceito	16
3.2	Comunicação GSM	18
3.2.1	O Terminal MC35i	19
3.3	Estrutura Base de Dados	20
3.3.1	Modelos de base de dados	21
3.3.2	Transacções	21
3.3.3	SQL	22

4	Arquitectura do projecto	23
4.1	Arquitectura do Projecto	24
4.2	A aplicação	25
4.2.1	Graphical User Interface	26
4.2.2	Data Acquisition	27
4.2.3	GSM State Machine	29
4.2.4	Máquina de Estados de Despacho de Notificações	34
4.3	Gestão de utilizadores	36
4.3.1	Gestão de acessos	37
4.3.2	A tabela de piquete	37
4.4	Telemanutenção	38
5	Apresentação de resultados	39
5.1	Apresentação dos diferentes módulos	40
5.2	O Interface	40
5.2.1	Tratamento Notificações	40
5.2.2	Módulo de Pesquisa	42
5.2.3	Filtros de Alarmes	43
5.2.4	Configuração da porta série	43
5.3	Os utilizadores	44
5.3.1	Novo utilizador	44
5.3.2	Tabela de Piquete	44
5.3.3	Histórico de Utilizações	46
5.4	Limitações do Sistema	47
5.5	Sistema Telemanutenção Industrial	48
6	Conclusões	51
6.1	Discussão geral dos resultados	52
6.2	Trabalhos futuros	53
7	Manual do Utilizador	57
7.1	Início	58
7.2	Menu de Sistema	58
7.3	Écran Principal	59
7.4	Pesquisa de Alarmes	59
7.5	Chamadas Efectuadas	60
7.6	Configuração de Alarmes	60
7.7	Configurações do Modem GSM	61
7.8	Gestão de Acessos	62

7.8.1	Novo Utilizador	62
7.8.2	Edição de Utilizador	64
7.8.3	Actividade dos Utilizadores	64
7.8.4	Tabela de Piquete	65
7.9	Sair da Aplicação	65
8	Código Fonte	67
	Bibliografia	73

Lista de Figuras

1 Introdução	1
1.1 Diagrama de uma empresa ilustrando várias secções de produção	3
1.2 Diagrama de blocos muito simples de um PLC	3
1.3 Exemplo de um PLC (Siemens S7-200)	4
1.4 Encadeamento do processo	4
1.5 Estrutura base do Servidor	6
1.6 Estrutura base do Cliente	6
1.7 Organização do Projecto	6
 2 Bases de Trabalho	 9
2.1 Desenho final (simplificado) do projecto	11
2.2 Mitsubishi MAM	12
2.3 Insys GSM	12
 3 Componentes do sistema	 13
3.1 Pormenor do ambiente de trabalho em LabVIEW (front panel e block diagram) 14	
3.2 Exemplo simples de código em LabVIEW para efectuar a equação 3.1	15
3.3 Icon de um vi	16
3.4 Exemplo simples de código em LabVIEW para ler dados da porta série	17
3.5 Exemplo simples de código em LabVIEW para efectuar um select à tabela_registos 17	
3.6 Logotipo GSM	18
3.7 Terminal Siemens MC35i	19
 4 Arquitectura do projecto	 23
4.1 Exemplo de de uma unidade fabril	24
4.2 Telemanutenção de Alarmes	25
4.3 Estrutura base do projecto	26
4.4 Exemplos de Alarmes em uma aplicação de supervisão local	28

4.5	Acesso do STI ao diferentes sistemas locais	28
4.6	Fluxograma de verificação de alarmes (Novo Alarme/Resolução de Alarme) . .	30
4.7	Estrutura da base de dados	32
4.8	Fluxograma com a estrutura da máquina de estados de notificações	33
4.9	<i>Buffers</i> e a forma como se interligam com a máquina de estados	34
4.10	Fluxograma com a estrutura da máquina de estados despacho de chamadas/SMS	35
5	Apresentação de resultados	39
5.1	Inicialização da aplicação	40
5.2	Tabela com alarmes activos	41
5.3	Tabela com as notificações efectuadas para o alarme seleccionado	41
5.4	Tabela de histórico de alarmes da fábrica	42
5.5	Tabela de configuração de Alarmes	43
5.6	Tabela de configuração do Modem	44
5.7	Tabela de Piquete	45
5.8	Tabela de Pesquisa de Utilizadores	46
5.9	Login no sistema	48
5.10	Configuração da tabela de Piquete	48
5.11	Informações do sistema	49
5.12	Configuração do <i>modem</i> GSM	49
	Manual do Utilizador	57
7.1	Inicialização da aplicação	58
7.2	Menu do Sistema	58
7.3	Tabela de histórico de alarmes	59
7.4	Tabela de notificações efectuadas	60
7.5	Configuração de Alarmes	61
7.6	Tabela de configuração do Modem GSM	61
7.7	Tabela de Gestão de Logins	62
7.8	Logins	63
7.9	Novo Utilizador	63
7.10	Edição de Utilizador	64
7.11	Tabela de Piquete	65
	Código Fonte	67
8.1	main.vi	68
8.2	Controlo de interface com o utilizador	69
8.3	Evento de Login	70

8.4	Evento de Logout	70
8.5	Evento que mostra os alarmes activos	70
8.6	Configurações do módulo	71
8.7	Informações da Selmatron	71
8.8	Máquina de Estados GSM	71
8.9	Aquisição de dados de cada uma das supervisões da fábrica	72

Lista de Tabelas

4	Arquitectura do projecto	23
4.1	Unidade de Alarme	29
4.2	Tabela de Alarmes, e tabela de Chamadas	31

Lista de Acrónimos

STI	Sistema de Telemanutenção Industrial
PC	<i>Personal Computer</i>
GSM	<i>Global System for Mobile communication</i>
LAN	<i>Local Area Network</i>
GUI	<i>Graphical Users Interface</i>
API	<i>Application Programming Interface</i>
LabVIEW	<i>Laboratory Virtual Instrument Engineering Workbench</i>
PLC	<i>Programmable Logical Controller</i>
SCADA	<i>Supervisory Control And Data Acquisition</i>
SMS	<i>Short Message Service</i>
M2M	Machine to Machine
VI	<i>Virtual Instrument</i>
FPGA	<i>Field-Programmable Gate Array</i>
NI	<i>National Instruments</i>
VISA	<i>Virtual Instrument Software Architecture</i>
GPIB	<i>General Purpose Interface Bus</i>
PXI	<i>PCI eXtensions for Instrumentation</i>
SQL	<i>Structured Query Language</i>
ACID	<i>Atomicity, Consistency, Isolation, Durability</i>
DML	<i>Data Manipulation Language</i>

DDL *Data Definition Language*

DCL *Data Control Language*

DTL *Data Transact Language*

DQL *Data Query Language*

Capítulo 1

Introdução

Neste capítulo será feita uma apresentação de todo o projecto, sendo referidos os pontos mais importantes, o seu enquadramento, o objectivo principal que motivou este trabalho, a sua importância e a sua utilidade.

1.1 Identificação dos objectivos a atingir

Como o próprio nome do trabalho indica, este projecto visa o desenvolvimento de um sistema de Telemanutenção Industrial.

Neste caso específico trata-se de uma sistema com vários componentes desenvolvido para registar e gerir todos os alarmes reencaminhando mais importantes para um sistema de telecomunicações que informará os utilizadores que existe uma anomalia na fábrica numa determinada máquina.

Será ainda possível através deste sistema de através de um programa de acesso a Internet (*browser*), verificar o estado de uma unidade fabril.

Trata-se assim de um projecto que aborda um número significativo de áreas científicas, tais como a electrónica, o controlo, as ciências computacionais e problemas na área das telecomunicações.

1.2 Enquadramento global

Este projecto surge de uma parceria entre o Departamento de Electrónica e de Telecomunicações da Universidade de Aveiro e a Selmatron, Lda.

A Selmatron é uma empresa de automação industrial, que desenvolve soluções à medida e actua nos diferentes sectores da industria.

Tendo esta empresa um vasto conhecimento da realidade industrial, e das mais importantes necessidades dos clientes, que surge o conceito deste projecto.

Nas grandes fábricas, é uma necessidade constante a minimização de custos de produção. Um dos factores mais dispendioso para uma entidade empresarial, é existir uma anomalia em uma determinado sector, impedindo que a fábrica produza normalmente.

Devido ao variado tipo de infra-estruturas de produção existentes numa fabrica, é necessário a construção de um *software* suficientemente versátil para de forma a abranger todos os componentes passíveis de criarem um alarme que impeça a normal produção de uma fábrica.

Vejamos a seguinte configuração de uma fábrica (figura 1.1).

Existem várias zonas de produção, entre a preparação da matéria prima, até a parte de expedição.

Durante todos estes processos, existem vários obstáculos que podem impedir o normal funcionamento da fábrica. É sobre a análise destes obstáculos que este *software* se revela útil, uma vez que o operário da manutenção é informado, independentemente do local onde se encontra.

Com a informação prévia da anomalia, da sua natureza e da localização, é possível ao funcionário da manutenção apurar a sua causa e tomar as devidas precauções para resolver o problema.

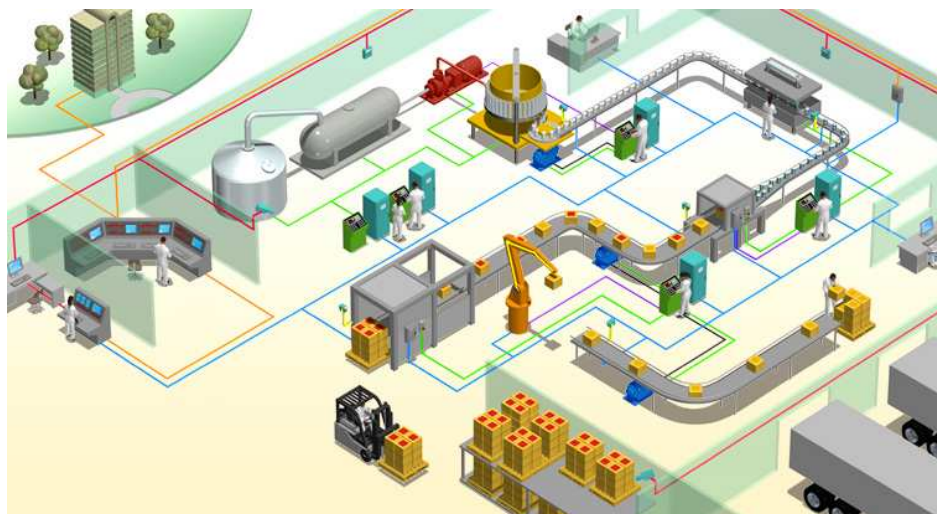


Figura 1.1: Diagrama de uma empresa ilustrando várias secções de produção

O conceito é bastante simples, um sistema que analise todos os alarmes de uma determinada entidade fabril, e que reencaminhe esses alarmes para um sistema de telecomunicações, com o fim de informar o pessoal responsável pela manutenção, da existência um problema e em que zona da fábrica.

Contudo existem inúmeros processos, variáveis e sistemas para supervisionar.

1.2.1 Alarmes

Esta secção procura explicar num processo industrial a noção de alarme e como é despoletado.

Existem várias realidades de processos industriais, da mesma forma que existem várias soluções.

No entanto, considere-se o um processo industrial genérico. Processo é controlado por um PLC, ou autómato, como é vulgarmente denominado.

De uma forma muito sintética um PLC, não é mais que um microcontrolador optimizado para controlo de processos e aquisição de sinal, em meio industrial.

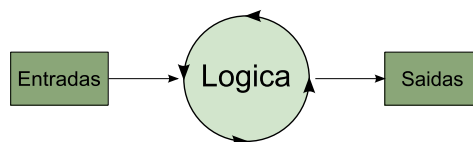


Figura 1.2: Diagrama de blocos muito simples de um PLC

Existem inúmeras configurações para os PLC's ao nível de processamento, protocolos de comunicação, entradas, saídas, robustez...



Figura 1.3: Exemplo de um PLC (Siemens S7-200)

Dependendo da filosofia do processo, o programador deve ter em atenção, que existem inúmeras condicionantes para que o processo seja interrompido, ou não seja correctamente executado (anomalias mecânicas ou eléctricas, falhas de processo, problemas de produção. . .).

Este ponto é particularmente importante neste projecto, pois são as notificações destes alarmes que são enviados para a secção de manutenção afim de serem rapidamente resolvidos.

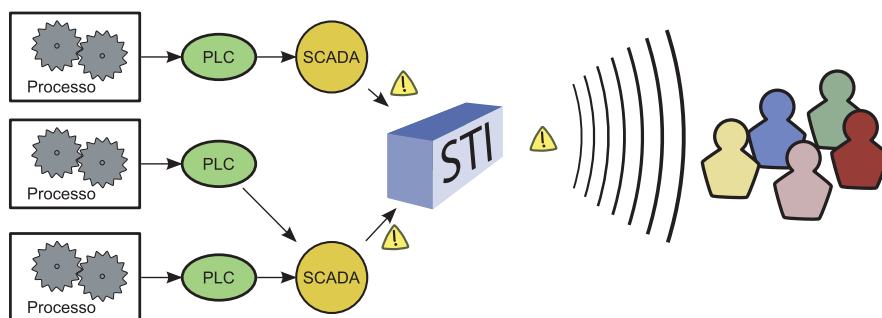


Figura 1.4: Encadeamento do processo

Consideremos a figura 1.4. Esta figura mostra a logística base deste projecto onde o PLC detecta as anomalias no processo, e as comunica com o SCADA (Unidade de supervisão local e/ou gestão de produção), por sua vez a supervisão comunica com uma unidade central de despacho de notificações, o STI.

Esta será a estrutura adoptada neste projecto.

1.3 Enquadramento Temporal

Este é o primeiro projecto resultante da parceria entre a Universidade de Aveiro através do Departamento de Electrónica e Telecomunicações e Informática e a empresa Selmatron, Lda. Surge no entanto no seguimento do projecto do ano anterior (2006/7) onde foi desenvolvido uma sistema de Telemantenção de Calandras. É importante referir, que o projecto actual em nada se assemelha ao anterior, quer ao nível da infra-estrutura de desenvolvimento, quer no resultado final.

O objectivo principal deste projecto é essencialmente criar uma forma mais rápida, produtiva e simples, de resolver avarias, reduzindo o tempo de máquina parada.

1.4 Objectivo base do projecto

De forma muito sumária o objectivo a atingir é o desenvolvimento de um sistema de telecomunicações que permita a comunicação com cada uma das máquinas, colocadas numa fábrica, permitindo a intervenção eficiente e mais rápida em caso de avaria.

Este sistema deve ser simples e versátil.

Simples para poder ser utilizado por várias pessoas com *culturas computacionais* diferentes.

Versátil para poder ser aplicável ao grande número de modelos de máquinas, sem que haja uma necessidade de alteração do código-fonte.

Deverá também ser independente do sistema de controlo do processo evitando problemas com a operação do programa de controlo e permitir a detecção de eventuais anomalias no sistema de controlo (falhas de comunicação, *por exemplo*).

O sistema deverá ter uma estrutura dividida em duas partes, separadas “geograficamente” isto é uma topologia Servidor Cliente.

Servidor

O servidor implantado numa unidade central (figura 1.5) deve estar em constante funcionamento para verificação e registo de todas as anomalias. Deve oferecer também a possibilidade de avaliar o estado instantâneo de alarmes de cada máquina e possuir os registos de todos os alarmes ocorridos em todas elas.

Cliente

A unidade cliente (figura 1.6) será apenas responsável pelo processo, e por a publicação de variáveis na rede. Todo o processamento relativo ao Sistema Telemantenção Industrial, fica reservado ao servidor.

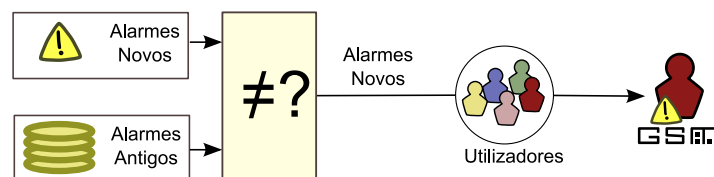


Figura 1.5: Estrutura base do Servidor

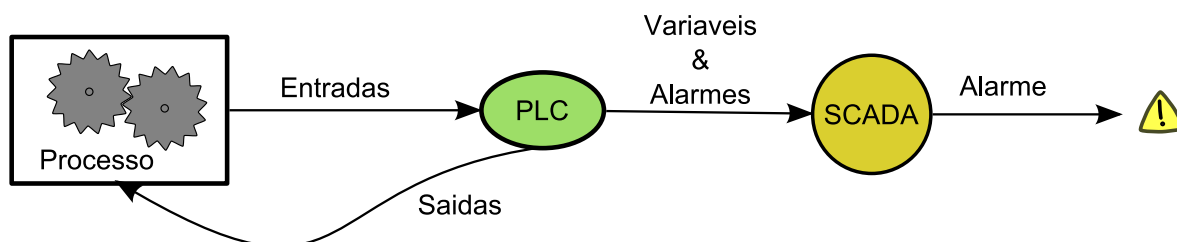


Figura 1.6: Estrutura base do Cliente

1.5 Importância do projecto

Este projecto oferece a oportunidade de redução de custos numa situação de paragem na produção (evitando o desperdício de matéria prima, desgaste anormal de equipamentos...).

Este sistema pode não conseguir antever nem tão-pouco resolver este tipo de situações. No entanto, informa, regista a comunicação, e o estado em que se descobriu o alarme.

O provável cliente desta aplicação terá uma mais valia na fábrica uma vez que com o apoio deste sistema uma vez que os tempo entre o acontecimento da anomalia e a sua resolução é diminuído significativamente.

1.6 Organização do projecto

Tendo em conta a grande dinâmica deste projecto e a sua complexidade, foi estruturado como se mostra no diagrama de blocos (figura 1.7):

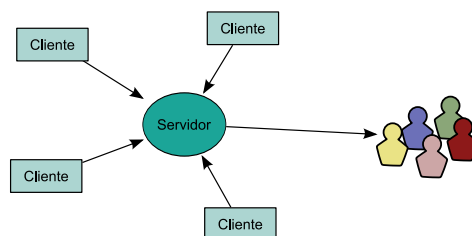


Figura 1.7: Organização do Projecto

Considerando o esquema da figura 1.7, deverá existir um **meio de comunicação** entre o

STI e os utilizadores que permita o envio de notificações, versátil e de baixo custo. Este meio de comunicação deverá ser implementado com suporte das infra-estruturas pré-existentes.

As notificações poderão ser feitas por meio de texto (SMS), ou de voz, e que o acesso a este tipo de informações não seja fixo.

O **sistema**, deverá ser fiável de modo a permitir a troca de informação de uma forma segura e eficaz entre o **servidor** e vários **clientes** (descritos na secção 1.4 na página 5).

1.7 Condições iniciais de Desenvolvimento

Tal como foi descrito anteriormente neste texto o desenvolvimento deste projecto foi fruto de uma parceria entre a Universidade de Aveiro, por intermédio do Departamento de Electrónica Telecomunicações e Informática, por a empresa de automação industrial Selmatron, Lda.

Esta parceria, criou uma conjectura que, de algum modo orientou o projecto a adoptar um caminho para a solução final, vejamos:

- A Selmatron, é actualmente integradora de sistemas National Instruments e Microsoft
- Pertence ao grupo Bresimar Automação SA, distribuidora de equipamentos para automação Siemens.
- O primeiro cliente da aplicação, foi a empresa Roca, SA.

Como seria de esperar, estes três pontos influenciaram em parte algumas das decisões a tomar no âmbito deste projecto. Contudo é importante referir que todos estes aspectos foram devidamente contemplados e discutidos ao longo de todas as fases do desenvolvimento do projecto.

Assim, vejamos quais foram as condições iniciais de desenvolvimento do projecto

1.7.1 O LabVIEW

O software de desenvolvimento adoptado foi o LabVIEW (da National Instruments). Para esta escolha contribuíram vários factores:

- O facto de a Selmatron ser integradora de sistema National Instruments.
- A orientação do LabVIEW, para programação de aplicações industriais.
- A compatibilidade e facilidade de integração com outros sistemas desenvolvido na mesma plataforma de desenvolvimento
- A robustez dos produtos da National Instruments, ao nível de *hardware* para comunicações industriais

Todos estes factores, influenciaram que no desenvolvimento deste projecto fosse utilizado o LabVIEW.

1.7.2 O *hardware*, e *software* de apoio

Ao nível de *hardware*, foi apenas utilizado um computador vulgar, que corria como sistema operativo o sistema Microsoft Windows XP, onde o *data-logging* era registado numa base de dados SQL Server, e onde a comunicação com os operários da manutenção era efectuada com um *modem* GSM da Siemens.

Estes factos, foram condicionados por diversos factores, tais como:

- A Selmatron ser integradora de sistemas Microsoft
- A fábrica da Roca, cliente final desta aplicação ter toda a sua estrutura de dados armazenada em base de dados SQL Server
- O facto de a Selmatron, fazer parte do grupo de empresas da Bresimar Automação, influenciou, a escolha relativamente ao *modem* GSM

Contudo todas estas abordagens serão descritas posteriormente neste texto.

Capítulo 2

Bases de Trabalho

Neste capítulo é apresentado um conjunto de possíveis soluções teóricas para o problema traçando já os caminhos a seguir para a sua concretização, desde a situação inicial, o *state of the art*, um *overview* de todas as componentes do sistema.

2.1 Técnicas de Comunicação

Após a descrição dos objectivos, é necessário delinear o caminho a percorrer para tal, assim tendo em conta o objectivo final deste projecto, existem vários caminhos possíveis para o atingir, onde cada uma das soluções envolve vantagens e desvantagens.

2.1.1 Via de Comunicação

Atentamos de novo a figura 1.7, onde podem ser identificados dois tipos de meios de comunicação necessários: Cliente → Servidor e Servidor → Utilizadores.

Neste trabalho serão avaliadas as seguintes opções:

- Ethernet *wired* (com fios) & *wireless* (sem fios)
- Rede Telefónica
- Rede GSM
- Rede 3G

Uma análise simples e tendo em conta o tráfego de dados que vai existir, parece óbvio que a via de comunicação entre o cliente e o servidor seja uma solução Ethernet, uma vez que o tráfego de dados será considerável e que esta solução já está adoptada nas fábricas ao nível de Gestão de Produção.

A título de curiosidade e conhecendo o meio industrial, a solução sem fio é rapidamente afastada, uma vez que são meios ricos em ruídos electromagnéticos derivado essencialmente de variadores de velocidade (*inverters*). Mesmo em soluções de cabo de cobre existem por vezes problemas de comunicação. Para resolução deste problema a solução passa por usar cabo de fibra óptica para construção desta rede, sendo esta solução completamente imune a este tipo de interferências.

2.1.2 Comunicação com os operários

Para a comunicação com os operários da manutenção o GSM revela-se a melhor escolha por ser uma solução simples, móvel, uma solução pouco dispendiosa, versátil.

Contudo tendo em conta que o tráfego de dados, resume-se a pequenas mensagens de texto, e chamadas de voz, esta solução cumpre perfeitamente os requisitos a que se propõe.

Assim o caminho tomado, pode ser descrito pela figura 2.1

2.2 *State of the art*

Da ideia deste projecto, surge um produto, algo que oferece vantagens a um cliente.

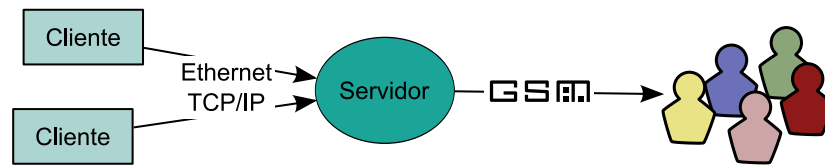


Figura 2.1: Desenho final (simplificado) do projecto

Após várias pesquisas ao longo do desenvolvimento do projecto, não foi encontrada nenhuma solução que se posicione ao mesmo nível deste trabalho.

Existem contudo, alguns produtos comerciais que podem oferecer algumas das potencialidades deste projecto.

2.2.1 Interfaces M2M

No mercado da automação industrial, a grande maioria das marcas deste sector oferece soluções de comunicação através de GSM (à semelhança do que foi utilizado neste trabalho Siemens MC35i¹), e outros protocolos de comunicação.

Serão apresentados alguns exemplos destes módulos.

Mitsubishi MAM

- Ligação RS232, RS485, RS422, para ligação directa ao PLC
- Envio de alarmes para um numero predefinido, SMS

Insys GSM

- Ligação RS232,
- Envio de alarmes para um numero predefinido, SMS
- Entradas e saídas programaveis

No entanto todos estes exemplos são apenas soluções orientadas para um único tipo de processo, em que não existe uma central de processamento de notificações para os encaminhar, quer ao nível de dois processos semelhantes, quer ao nível de uma fábrica com diferentes tipos de processo.

Com o STI, é possível otimizar o investimento ao nível de módulos e sua respectiva configuração, que não é trivial.

Este sistema não oferece esse tipo de preocupações, uma vez que todos os alarmes são reencaminhados para o servidor, e este gere o tratamento que é necessário aplicar.

¹Este módulo não será incluído nesta secção uma vez que será descrito com mais detalhe posteriormente neste trabalho

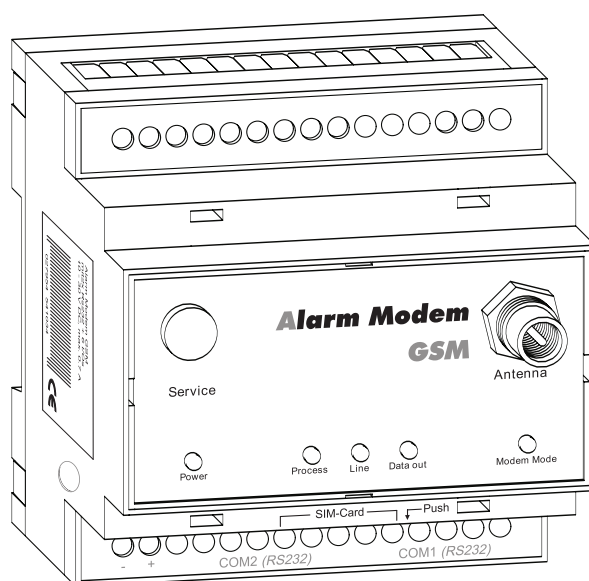


Figura 2.2: Mitsubishi MAM



Figura 2.3: Insys GSM

Capítulo 3

Componentes do Sistema

Este capítulo, é dedicado na sua totalidade à descrição de cada um dos componentes que constituem todo o sistema de telemanutenção desenvolvido.

Tendo em conta a natureza deste projecto e a diversidade de domínios científicos abrangidos, é de todo o interesse separar estes pontos.

Será aqui explicado cada um dos componentes desenvolvidos, os métodos de transmissão de informação, o tratamento dos dados, o registo de alarmes, todos os pequenos pormenores que compõem este projecto.

É também elaborada uma secção sobre a linguagem em que é desenvolvido o sistema responsável pelo tratamento de dados.

3.1 O LabVIEW

Nesta secção será feita uma breve descrição do software LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench). LabVIEW, é uma linguagem de programação proprietária da National Instruments.

Surgiu em 1986, e a programação é feita por meio de fluxo de dados, sendo considerada uma linguagem de programação gráfica.

É uma ferramenta essencialmente orientada para aquisição de dados, medições, instrumentação, controlo e processamento de sinal.

Cada unidade básica de programa criado em LabVIEW é denominado de Virtual Instrument (Instrumento Virtual) como se de uma ferramenta de laboratório se tratasse.

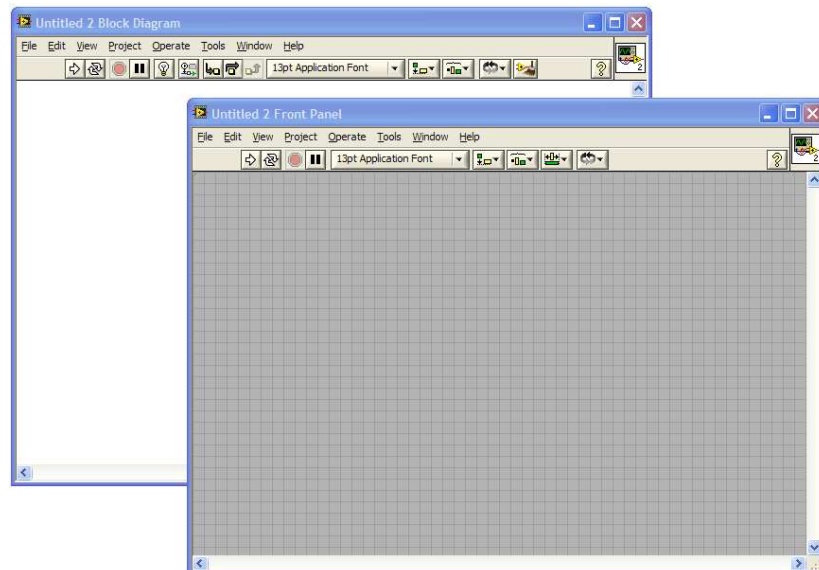


Figura 3.1: Pormenor do ambiente de trabalho em LabVIEW (front panel e block diagram)

Cada VI, é constituído por dois blocos distintos (figura 3.1), um painel frontal com o interface, e o diagrama de blocos, que contem o código do programa.

Todo o código em LabVIEW é compilado, visto que se trata de uma linguagem de alto nível, evitando assim o uso de um interpretador.

3.1.1 Programação por fluxo de dados

A linguagem de programação gráfica do LabVIEW é denominada G. A designação de G deriva de *Graphical programming*.

A execução do programa é determinada por uma estrutura gráfica organizada por blocos, facilmente se entende esta estrutura é denominada de código fonte de uma aplicação em LabVIEW. O programador, interliga os diferentes blocos com fios (*wires*) orientando assim o

fluxo de dados, como se tratasse de condutores eléctricos entre vários dispositivos electrónicos. Estes *wires*, propagam os valores presentes em memória (com valores variáveis ou constantes). Assim cada bloco que esteja ligado a um determinado *wire* vai usar esse valor.

Uma das mais importantes características deste tipo de programação é a filosofia intrínseca de *multi-threading* (execução de vários processos em paralelo), onde vários blocos podem ser executados ao mesmo tempo sem muitas preocupações para o programador. Este facto só é possível uma vez que o LabVIEW recorre directamente ao *scheduler* multiplexando as várias threads de modo a que o Sistema Operativo as consiga gerir.

Este conceito de programação pode se mostrar propício a *race-condition*, (isto é, variáveis com estado indefinido), caso não seja adoptada uma estrutura de programa do tipo, onde seja respeitado o fluxo dos dados. A título de curiosidade um dos princípios elementares da programação em LabVIEW é:

“Always respect the data-flow”

Vejamos um exemplo simples de programação em LabVIEW :

Consideremos a equação:

$$2\pi \times r = P \quad (3.1)$$

Em LabVIEW teríamos algo deste género:

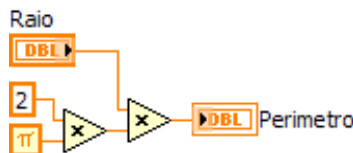


Figura 3.2: Exemplo simples de código em LabVIEW para efectuar a equação 3.1

Como é simples de entender o resultado.

3.1.2 Programação Gráfica

A programação gráfica é outro dos pontos fortes desta linguagem. Cada rotina em LabVIEW, é denominada de VI (Virtual Instrument), como se de um instrumento virtual se tratasse.

Cada VI é constituído por duas estruturas distintas (figura 3.1):

- Interface Gráfico para Utilizador (*front panel*)
- Diagrama de blocos (*block diagram*)

Relativamente às rotinas/subrotinas, é necessário revelar a forma como os parâmetros de entrada e saída são abordados. Cada **vi** deve ser sempre acompanhado de um *icon*, que identifique a sua função, este *icon* tem pontos de entrada e de saída configuráveis, vejamos:

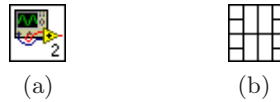


Figura 3.3: Icon de um **vi**

Cada uma das pequenas células estão disponíveis para serem configuradas como entradas ou saídas (com *wires*) de uma função que estará presente no diagrama de blocos.

Tal como em outras linguagens de programação o formato dos parâmetros de entrada pode ser qualquer, desde uma simples **boolean**, **integer**, **cluster**, **variant**,...

3.1.3 Blocos, o conceito

Como vimos anteriormente, o conceito de blocos, é a base de desta linguagem de programação.

A filosofia do bloco torna-se intrínseca ao LabVIEW uma vez que cada utilizador apenas usa os módulos que necessita.

O módulo base é constituído por o ambiente de programação LabVIEW o *run-time* e módulos de aquisição/geração/tratamento de sinal.

Contudo se um utilizador pretende fazer controlo em tempo-real temos o *Módulo Real Time + FPGA*, ou se prefere construir sistemas *embedded* para μ Controladores *Módulo Embedded*, ou até o *Módulo Vision* para sistemas de visão ... Existem bastantes blocos de funções adequados a áreas de interesse mais específicas.

É também importante referir que o LabVIEW é uma linguagem de programação multi-plataforma, com versões para os vários sistemas operativos, Windows, Mac OS, Linux. Contudo, apenas o módulo base do LabVIEW é multi-plataforma.

Existe ainda muito mais para falar sobre este software, no entanto alguns pormenores serão descritos ao longo deste texto quando assim se justificar.

É conveniente referir que sendo o LabVIEW uma linguagem proprietária da National Instruments, existem um conjunto de infra-estruturas previamente criadas de interface com o hardware na NI, Tornando a integração deste hardware com projectos desenvolvidos em LabVIEW, *painless*, isto é simplificando muito o projecto a nível de protocolos de comunicação, o desenvolvimento, condicionamento de sinal, calibrações, aplicação...

Neste trabalho será descrito com mais detalhe dos módulos VISA e Database Connectivity.

Módulo VISA

O módulo VISA (Virtual Instrument Software Architecture), é um standard para configuração, programação e resolução de problemas (*troubleshooting*) dos vários interfaces de comunicação disponíveis no *hardware* da National Instruments. O interface série é um exemplo integrado no pacote VISA. Tratam-se assim de drivers previamente desenvolvidos para efectuar comunicações com a porta série.

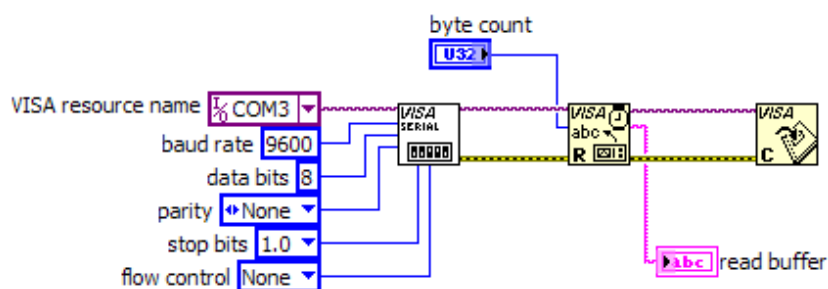


Figura 3.4: Exemplo simples de código em LabVIEW para ler dados da porta série

Este módulo é bastante importante, para a comunicação com o módulo GSM, comunicando com este através de comandos AT.

Módulo Database Connectivity

Este módulo como o próprio nome indica, é o responsável por a comunicação com a base de dados desenvolvida.

Consiste em vários blocos com as funções mais utilizadas de acesso a base de dados.

O acesso é feito através de bibliotecas internas do Windows.

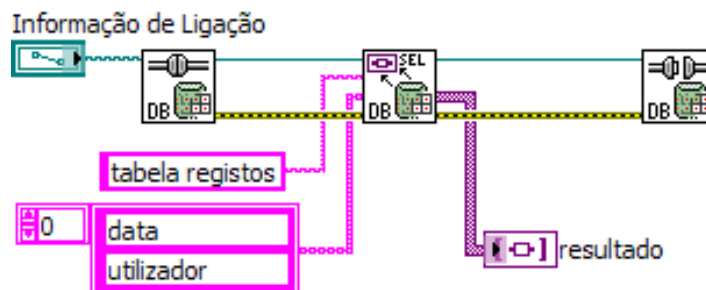


Figura 3.5: Exemplo simples de código em LabVIEW para efectuar um select à tabela _registos

3.2 Comunicação GSM

GSM, *Global System for Mobile Communications*, (GSM: originalmente, Groupe Spécial Mobile) é uma tecnologia móvel e o padrão mais popular nos telefones móveis em todo o mundo.

Distinguiu-se dos seus antecessores, por o canal de voz, ser um sinal digital e por a comunicação de dados ser contemplada desde o início da criação deste protocolo.

Trata-se de um padrão de telecomunicações aberto, tendo sido desenvolvido pela 3GPP



Figura 3.6: Logotipo GSM

O GSM possui uma série de características que o distinguem dentro do universo das comunicações móveis. Nascido nos anos 80 é fruto de uma cooperação entre várias entidades Europeias, o sistema partilha elementos comuns com outras tecnologias utilizadas em telemóveis, tais como:

- Transmissão digital
- Infra-estrutura celular

Serão de seguida apresentadas as características fundamentais do sistema, assim como as suas capacidades, de uma forma muito sintética.

Do ponto de vista do consumidor, a vantagem-chave do GSM é o novo conceito de comunicação, a preços tentadores. O SMS, foi um destes novos conceitos.

Ao nível das operadoras o baixo custo das infra-estruturas deve-se à grande oferta e o facto de se ter tornado um modo de comunicação de massas, torna-o muito *apetecido*.

A principal desvantagem é que o sistema GSM é baseado na rede TDMA que ao contrário do CDMA, atribui *pacotes* de tempo para comunicação (enquanto que o CDMA atribui frequências). Percebe-se facilmente que não permite comunicação em dois blocos de tempo em simultâneos. Implica assim que caso todas as células disponíveis para comunicação estejam ocupadas, a ligação vai cair no fim do *slot* de tempo.

Posteriormente a transmissão de dados em alta velocidade foi adicionada no novo esquema de modulação EDGE. A versão de 1999 deste padrão introduziu índices de transmissão de dados altos, e é normalmente referida como 3G.

A comunicação GSM, é uma tecnologia relativamente recente, pelo que veio revolucionar os meios de comunicação da sociedade do século XX.

Como foi anteriormente referido neste documento, o principal meio de comunicação do sistema para envio de notificações foi o GSM, por ser bastante versátil e uma solução relativamente económica.

3.2.1 O Terminal MC35i

O módulo Siemens MC35i é um terminal GSM/GPRS, desenvolvido para o meio industrial.



Figura 3.7: Terminal Siemens MC35i

Este módulo permite efectuar ligações GSM, que o torna a solução ideal para este projecto. Consegue-se assim duas fortes vantagens face a um sistema de ligação com fios.

- A facilidade de comunicação com um sistema computacional através da porta série, para funções típicas de sistemas de comunicação GSM, tais como as SMS.
- A portabilidade do sistema e a possibilidade de ser aplicado em unidades colocadas em locais distantes e inóspitos (*levantando um pouco o véu sobre possíveis trabalhos futuros*)

Para comunicar com o módulo serão utilizados comandos AT, enviados através da porta série RS232.

Comandos AT

Os comandos AT são também denominados de conjunto de comandos de Modem Hayes.

Os modems Hayes foram durante muito anos um padrão na tecnologia dos modems. Toda-via com o crescimento tecnológico a produção de modems em massificou-se, e vários fabricantes aderiram aos comandos AT (*salvo alguns comandos que variam entre fabricantes*).

Assim os comandos AT podem ser divididos em quatro grupos distintos:

Comandos Básicos Um carácter maiúsculo seguido de um número.

Comandos Extendidos Um & seguido de Comando básico

Comandos Proprietários Vulgarmente iniciados por \ ou por %, comandos estes que variam entre fabricantes.

Comandos de Registos Operações a nível interno com a memória do modem

O funcionamento é bastante simples, os comandos são enviados para o modem e o modem responde com mensagens que permitem ter *feedback* do estado em que se encontra, e saber o resultado da acção do comando.

Vejamos um pequeno exemplo de marcação de uma chamada com comandos AT.

```
1 ATD+351912345678
2 >OK
```

Desta forma é iniciada a chamada de voz para o numero de telefone

+351 912 345 678

Para terminar a chamada, basta enviar para o modem:

```
1 ATH
2 >OK
```

E este termina todas a chamadas activas nesse momento.

3.3 Estrutura Base de Dados

De uma forma muito simplista uma base de dados é uma estrutura de dados que agrupa informação de uma forma organizada, para que os registos guardados sejam utilizados para o mesmo fim.

Existem várias estruturas de base de dados, contudo o modelo mais utilizado é o modelo onde a informação se encontra estruturada por tabelas, compostas por linhas e colunas.

As bases de dados são particularmente importantes em utilizações com múltiplos utilizadores, em que é imperativo a coordenação de vários utilizadores, com vários tipos de permissões. São também úteis para gestão de dados de utilizadores individuais, por a informação se encontrar de forma organizada e por a pesquisa de informação ser relativamente rápida.

3.3.1 Modelos de base de dados

Existem vários modelos de base de dados.

O modelo plano consiste numa estrutura de dados organizada por matrizes simples, bidimensionais, compostas por elementos de dados: inteiros, números reais, etc.

O modelo em rede permite que várias tabelas sejam usadas simultaneamente através do uso de ponteiros (ou referências). Este modelo é organizado por uma estrutura de grafos. Algumas colunas, em vez de dados, podem ter ponteiros para outras tabelas. Assim, as tabelas são ligadas por referências, o que pode ser visto como uma rede. Uma variação particular deste modelo em rede, é o modelo hierárquico, que limita as relações a uma estrutura semelhante a uma árvore.

Bases de dados relacionais consistem, principalmente de três componentes:

coleção de estruturas de dados - relações, tabelas;

coleção dos operadores , a álgebra e o cálculos relacionais;

coleção de restrições da integridade , definindo o conjunto consistente de estados de base de dados e de alterações de estados.

As bases de dados relacionais permitem aos utilizadores escreverem consultas (*queries*) que não foram previstas por quem projectou a base de dados. Como resultado, bases de dados relacionais podem ser utilizadas por várias aplicações em formas que os projectistas originais não previram, o que é bastante importante em bases de dados que podem ser utilizadas durante décadas. Isto tem tornado as bases de dados relacionais muito populares no meio empresarial.

3.3.2 Transacções

O conjunto de procedimentos que é executado numa base de dados, é denominado de transacção, que o utilizador considera como uma única acção. A integridade de uma transacção depende de quatro propriedades conhecidas como ACID.

Atomicidade Todas as acções que constroem uma transacção devem ser completadas com sucesso, para que a operação seja executada. Qualquer operação que implique uma falha na transacção deve ser desfeita (*rollback*). Quando todas as operações forem executadas correctamente a transacção deve ser efectivada (*commit*).

Consistência Nenhuma operação em base de dados deve ser parcial. O estado de uma transacção deve ser completado na íntegra.

Isolamento Cada transacção é independente de qualquer outra. Nenhuma outra transacção que seja executada no mesmo sistema, pode interferir com o funcionamento da transacção corrente. Da mesma forma outras transacções não podem visualizar os resultados parciais das operações da transacção corrente.

Durabilidade Este conceito implica que os resultados de uma transacção sejam permanentes e só podem ser desfeitos por uma transacção subsequente.

3.3.3 SQL

O SQL, é uma linguagem de pesquisa declarativa para uma base de dados relacional.

Foi desenvolvida para demonstrar a viabilidade de implementação de modelos relacionais de base de dados, baseados em álgebra relacional.

É actualmente um dos maiores *standards* da linguagem de base de dados, muito graças a sua simplicidade e facilidade de uso, uma vez que a consulta especifica a forma do resultado, e não a forma de o obter.

Palavras chave em SQL

Em SQL, existem uma série de comandos, agrupados da seguinte forma:

DML - Linguagem de Manipulação de Dados

DDL - Linguagem de Definição de Dados

DCL - Linguagem de Controle de Dados

DTL - Linguagem de Transacção de Dados

DQL - Linguagem de Consulta de Dados

Destes pequenos grupos de linguagens, apenas será referido o DML, por ser o mais utilizado.

SELECT é o comando mais usado do SQL, que permite ao utilizador especificar o formato do resultado desejado.

INSERT é utilizado para inserir uma linha em uma tabela já existente de uma base de dados.

UPDATE é usado para alterar os valores de uma determinada posição numa linha de uma tabela existente.

DELETE é o comando que permite apagar uma linha de uma dada tabela da base de dados.

O SQL representa assim uma mais valia importante neste projecto que onde são garantidos a segurança dos dados, e a organização de toda a estrutura.

Assim conclui-se esta sintética descrição das peças fundamentais na elaboração deste trabalho.

Todo o software foi desenvolvido em LabVIEW, e toda a estrutura referente a *debugging* da base de dados foi feita em Microsoft SQL Server, ambas as licenças de software são proprietárias da empresa Selmatron, Lda.

Capítulo 4

Arquitectura do projecto

Este capítulo, é um dos mais importantes deste texto, pois engloba toda a descrição da arquitectura do trabalho executado desde o processamento de dados até à transmissão de informação, passando também pela comunicação necessária com os vários programas de controlo.

O texto será acompanhado de algum código e de alguns fluxogramas que apoiam a compreensão, quer a abordagem tomada quer do método de funcionamento dos diferentes blocos.

O desenvolvimento do capítulo será feito por níveis que indicam quer a complexidade a nível da organização do projecto.

É importante referir, que o projecto desviou-se do seu objectivo inicial, uma vez que inúmeras funcionalidades foram adicionadas que o tornam numa mais valia.

4.1 Arquitectura do Projecto

Nesta secção será abordada a estrutura que foi adoptada na elaboração deste projecto, a numa primeira fase a nível de equipamento que foi necessário, assim como das diferentes abordagens ao nível do desenvolvimento computacional.

Tendo em conta uma estrutura de uma fábrica, onde cada vez mais a tendência é criar linhas de comunicação entre os *escritórios* e a *produção*, essa mesma estrutura de comunicação pode todavia ser útil para a utilização deste sistema de Telemantenção Industrial, onde é necessário que se garanta a conectividade a todos os departamentos de produção da entidade fabril. Atentamos assim ao diagrama 4.1

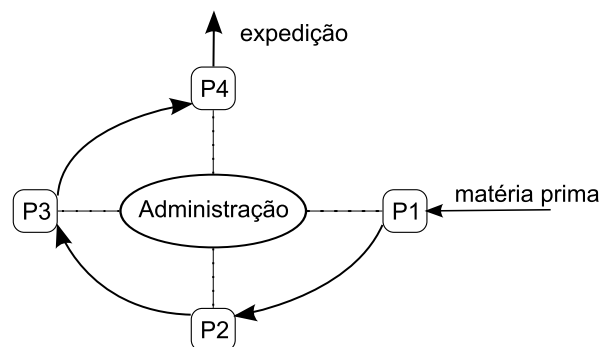


Figura 4.1: Exemplo de de uma unidade fabril

As linhas a cheio representam o trajecto do produto, enquanto que as linhas a tracejado correspondem a infra-estrutura de comunicações montada para a gestão de produção por parte da administração.

Consideremos assim uma unidade fabril da empresa XPTO, SA com as unidades de produção:

- Secção P1 - Materia Prima
- Secção P2 - Processo 1
- Secção P3 - Processo 2
- Secção P4 - Expedição

Cada uma destas secções controlam pequenos processos que transformam a matéria prima no produto final ao longo de todo o ciclo $P1 \rightarrow P2 \rightarrow P3 \rightarrow P4$. Cada uma das Secções tem vários módulos de controlo, *supervisionados* por um sistema computacional, que tem várias funções:

- Interface Utilizador

- Histórico de Registos
- Informações de Produção

Todas as unidades computacionais das diferentes estações de produção. No entanto todas estas secções estão ligadas fisicamente à **Secção A** por uma rede de comunicação (LAN) a partir da qual é gerida toda a produção fabril.

Esta é a estrutura alvo do Sistema de Telemantenção Industrial. Várias unidades de produção, como processos encadeados onde os atrasos, devem ser controlados a fim de evitar lapsos na produção.

Assim, com uma unidade (na **Secção A**), a supervisionar todos os departamentos de produção, verificando a existência de alarmes nos diferentes pontos de processo que impeçam o normal funcionamento da fábrica. A secção de Manutenção, seria assim avisada onde e quando ocorreu o alarme, podendo-o resolver mais rapidamente.

Este trabalho permite ainda que a Selmatron, possa remotamente monitorizar esta aplicação, de modo a poder resolver à distância eventuais problemas.

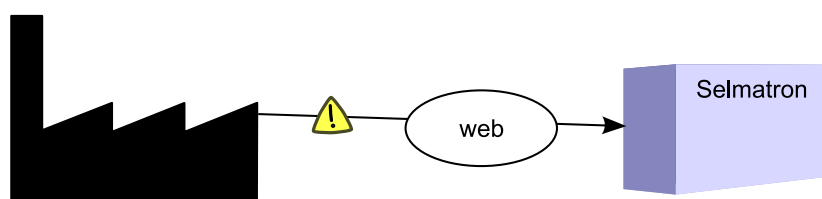


Figura 4.2: Telemantenção de Alarmes

Esta é estrutura genérica em uma fábrica de transformação, tornando-se assim o principal alvo *eventual* deste projecto, tendo em conta uma perspectiva comercial.

4.2 A aplicação

O sistema desenvolvido no âmbito deste projecto visa ser uma ferramenta de apoio, à produção de uma fábrica genérica, tendo que ser extremamente versátil para obedecer as exigências do mercado industrial. Esse apoio pode representar um aumento de produtividade de uma forma indirecta, no entanto este é um tema para ser desenvolvido mais adiante neste documento.

Várias abordagens seriam possíveis de realizar com o objectivo que foi proposto, como foi visto anteriormente no *state of the art*, todavia o Sistema de Telemantenção Industrial, não pretende ser uma caixa preta que se limita a informar que existe uma anomalia. O conhecimento adquirido com o desenvolvimento deste trabalho aponta para uma possível integração para além do âmbito académico, o que motiva este trabalho a possuir uma estrutura

com alguma interactividade e funcionalidades impossíveis de conseguir com um sistema mais simples.

A estrutura geral do STI pode ser representada da forma como é demonstrada na figura 4.3.

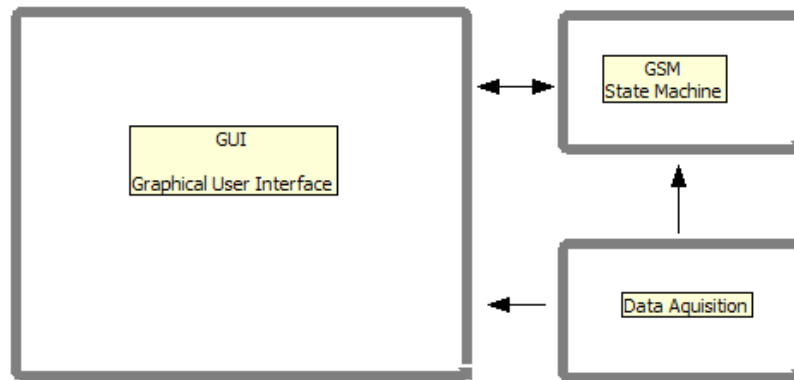


Figura 4.3: Estrutura base do projecto

Existem assim três blocos fundamentais:

1. GUI - Graphical User Interface (Interface gráfico com utilizador)
2. GSM Sate Machine - Máquina estados módulos GSM
3. Data Acquisition - Aquisição de dados

4.2.1 Graphical User Interface

Por se tratar de um projecto que se procura distinguir por responder a uma série de necessidades, oferece assim várias potencialidades.

Após alguns estudos junto de várias entidades fabris, organizou-se uma série de pontos que seriam relevantes acrescentar ao projecto, de forma a torná-lo uma mais valia e a procurar responder as necessidades de quem necessita de um sistema deste tipo.

A aplicação deve permitir:

- Visualizar todos os alarmes activos na fábrica;
- Ter um sistema de gestão de acessos;
- Registar em base de dados SQL, todos os alarmes da fabrica;
- Registar todos os eventos que foram reencaminhados para os utilizadores, isto é, cada mensagem que foi enviada, cada chamada efectuada
- Ter uma lista de configuração de alarmes.

- Permitir acesso remoto

Esta aplicação deverá ser executada num sistema computacional que deverá possuir uma interface de comunicação série RS-232, de forma a ser possível a comunicação com o módulo GSM MC35i.

Para além da do interface gráfico a aplicação deve gerir a comunicação com todos os pontos remotos e com o módulo GSM. A fim de garantir a comunicação com os utilizadores em caso de alarme.

Mais a frente neste trabalho serão descritas as abordagens que foram adoptadas para cada um dos pontos que foram referidos relativamente à aplicação do computador.

4.2.2 Data Acquisition

Este módulo é mais um dos módulos chave deste projecto uma vez que sem ele não seria possível ter toda a informação dos alarmes da fábrica centralizados numa só aplicação.

A aquisição de dados consiste basicamente numa, estrutura que corre toadas as aplicações de supervisão de processo e procura os alarmes que estas tem disponíveis.

Clientes

Cada uma das unidades de supervisão dos pontos P1, P2, P3 e P4 da figura 4.1 disponibiliza na rede interna da fábrica os alarmes que tem activos no sistema. Quando ocorre um alarme a aplicação local¹ regista internamente o alarme, notifica o utilizador através de informações visuais no interface (figura 4.4, página 28).

O STI tem que, em tempo real, verificar todos os alarmes da fábrica, e actuar conforme as configurações que foram definidas. A estrutura simplificada pode ser traduzida como indicado na figura 4.5, página 28.

Cada uma das aplicações remotas publica a sua lista de alarmes no seu IP, num protocolo previamente definido e disponibilizado pelo LabVIEW (de modo a não ser corrompido por terceiros). É disponibilizada uma lista com unidades de alarme com uma série de informações relativa a este. Estas unidades são do tipo (tabela 4.1):

Cada alarme que acontece tem um cluster deste tipo associado, e cada supervisão tem uma lista com todos os alarmes activos.

Torna-se necessário haver apenas o processamento do lado do cliente, suficiente para preencher dos campos para o fornecimento de informações ao Sistema de Telemanutenção Industrial.

Servidor

O STI, como demonstrado na figura 4.5, através da rede de comunicação física, acede a cada um dos clientes onde lê a sua lista de alarmes. É assim iniciado um processo crucial neste

¹Entenda-se aplicação local cada uma das supervisões de processo *dispersas* pela fábrica

Alarmes activos							
Descrição	Zona	Setpoint	Valor	Início	Fim	Reconhecimento	Utilizador
Botão Emergência	Linha 12	0	0	15-07-2008 15:51:46		15-07-2008 15:52:11	admin
Botão Linha em Manual	Linha 12	0	0	15-07-2008 15:51:46		15-07-2008 15:52:11	admin
Botão Emergência	Linha 16	0	0	15-07-2008 15:27:50		15-07-2008 15:28:59	admin
Botão Linha em Manual	Linha 16	0	0	15-07-2008 15:27:50		15-07-2008 15:28:59	admin
Botão Emergência	Linha 17	0	0	15-07-2008 15:27:50		15-07-2008 15:28:59	admin
Botão Linha em Manual	Linha 17	0	0	15-07-2008 15:27:50		15-07-2008 15:28:59	admin
Botão Emergência	Linha 20	0	0	15-07-2008 15:27:50		15-07-2008 15:28:59	admin
Botão Linha em Manual	Linha 20	0	0	15-07-2008 15:27:50		15-07-2008 15:28:59	admin
Botão Emergência	Linha 71	0	0	15-07-2008 15:27:50		15-07-2008 15:28:59	admin
Botão Linha em Manual	Linha 71	0	0	15-07-2008 15:27:50		15-07-2008 15:28:59	admin
Botão Emergência	Linha 81	0	0	15-07-2008 15:27:50		15-07-2008 15:28:59	admin
Botão Linha em Manual	Linha 81	0	0	15-07-2008 15:27:50		15-07-2008 15:28:59	admin
Botão Emergência	Linha 2	0	0	15-07-2008 15:27:49		15-07-2008 15:28:59	admin
Botão Linha em Manual	Linha 2	0	0	15-07-2008 15:27:49		15-07-2008 15:28:59	admin
Botão Emergência	Linha 3	0	0	15-07-2008 15:27:49		15-07-2008 15:28:59	admin
Botão Linha em Manual	Linha 3	0	0	15-07-2008 15:27:49		15-07-2008 15:28:59	admin
Botão Emergência	Linha 4	0	0	15-07-2008 15:27:49		15-07-2008 15:28:59	admin
Botão Linha em Manual	Linha 4	0	0	15-07-2008 15:27:49		15-07-2008 15:28:59	admin
Botão Emergência	Linha 5	0	0	15-07-2008 15:27:49		15-07-2008 15:28:59	admin
Botão Linha em Manual	Linha 5	0	0	15-07-2008 15:27:49		15-07-2008 15:28:59	admin
Botão Emergência	Linha 6	0	0	15-07-2008 15:27:49		15-07-2008 15:28:59	admin
Botão Linha em Manual	Linha 6	0	0	15-07-2008 15:27:49		15-07-2008 15:28:59	admin
Botão Emergência	Linha 7	0	0	15-07-2008 15:27:49		15-07-2008 15:28:59	admin
Botão Linha em Manual	Linha 7	0	0	15-07-2008 15:27:49		15-07-2008 15:28:59	admin
Botão Emergência	Linha 8	0	0	15-07-2008 15:27:49		15-07-2008 15:28:59	admin
Botão Linha em Manual	Linha 8	0	0	15-07-2008 15:27:49		15-07-2008 15:28:59	admin
Botão Emergência	Linha 9	0	0	15-07-2008 15:27:49		15-07-2008 15:28:59	admin
Botão Linha em Manual	Linha 9	0	0	15-07-2008 15:27:49		15-07-2008 15:28:59	admin
Botão Emergência	Linha 10	0	0	15-07-2008 15:27:49		15-07-2008 15:28:59	admin
Botão Linha em Manual	Linha 10	0	0	15-07-2008 15:27:49		15-07-2008 15:28:59	admin
Botão Emergência	Linha 11	0	0	15-07-2008 15:27:49		15-07-2008 15:28:59	admin
Botão Linha em Manual	Linha 11	0	0	15-07-2008 15:27:49		15-07-2008 15:28:59	admin
Botão Emergência	Linha 2	0	0	15-07-2008 15:27:48		15-07-2008 15:28:59	admin

Figura 4.4: Exemplos de Alarmes em uma aplicação de supervisão local

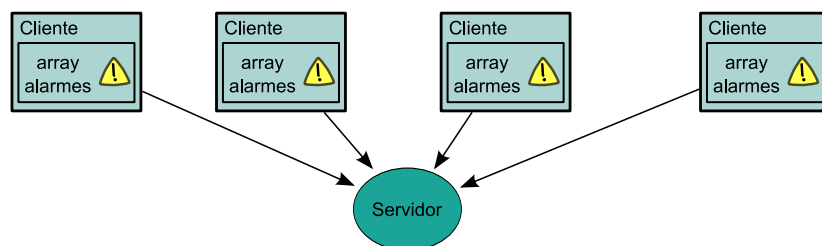


Figura 4.5: Acesso do STI ao diferentes sistemas locais

<code>alarm URL (string)</code>	Cliente onde o alarme está activo
<code>set time (time stamp)</code>	Hora de inicio do alarme
<code>set user (string)</code>	Utilizador no inicio do alarme
<code>ack time (time stamp)</code>	Hora de reconhecimento do alarme
<code>ack user (string)</code>	Utilizador do reconhecimento
<code>clear time (time stamp)</code>	Hora de resolução do alarme
<code>clear user (string)</code>	Utilizador da resolução
<code>setpoint (double)</code>	Setpoint do alarme
<code>value (double)</code>	Valor do alarme
<code>priority (integer 32)</code>	Prioridade do alarme
<code>alarm area (string)</code>	Area do alarme
<code>alarm description (string)</code>	Descrição do alarme
<code>ack comment (string)</code>	Comentário de reconhecimento
<code>event? (bool)</code>	Filtro de alarmes
<code>refnum (unsigned 64)</code>	Identificador de alarme

Tabela 4.1: Unidade de Alarme

sistema de gestão representado na figura 4.6.

Este processo está incluído num dos módulos que é executado de forma independente para cada um dos clientes.

Aqui beneficia-se de duas das propriedades mais populares das bases de dados:

- operações de acesso atómicas (leitura e escrita);
- velocidade de pesquisa.

Neste processo os alarmes são separados em duas classes, alarmes novos, e alarmes resolvidos. Quando um alarme é novo, é feita uma pesquisa prévia a base de dados, e regista-o na base de dados caso seja um novo alarme. Caso o alarme já exista na base de dados e entretanto desaparecer do array de alarmes disponível no cliente, significa que o alarme foi resolvido, é então feito um update a situação do alarmes com os novos parâmetros (tais como, `ack time`, `ack user`, `clear time`, `clear user`).

Desta forma, é construída uma base de dados que suporta todo o Sistema de Telemanutenção Industrial com a estrutura que será apresentada na figura 4.7

Tendo o registo dos alarmes no servidor, falta agora referir toda a estrutura responsável pelo envio de notificações via GSM aos utilizadores.

4.2.3 GSM State Machine

Esta secção será descrever a estrutura que controla todo o processo de notificações do módulo GSM, desde o envio de chamadas desde à escolha do utilizador a ser notificado até ao reenvio de chamada em caso de falha.

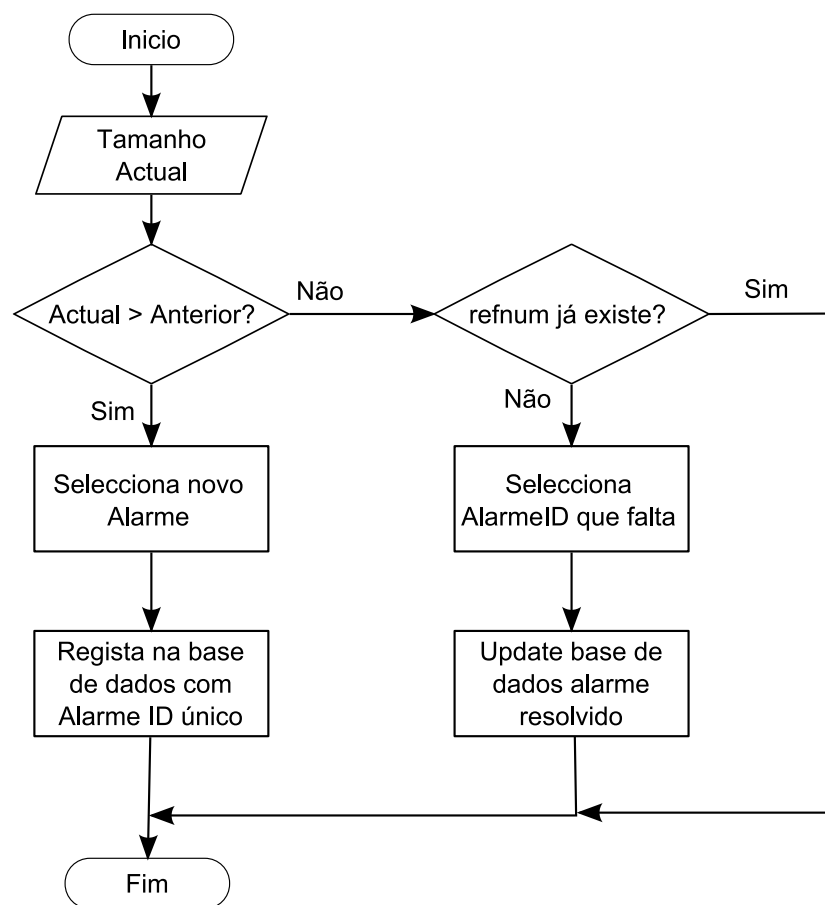


Figura 4.6: Fluxograma de verificação de alarmes (Novo Alarme/Resolução de Alarme)

Estrutura do sistema

O seguinte *layout* (tabela 4.2), corresponde a estrutura criada para registo do alarmes e das notificações tratadas, que por sua vez serve de sincronismo ao sistema de notificações. Nesta fase será apenas, por motivo de simplificação, referida a parte da estrutura de base de dados que se revela fundamental neste estágio.

Existem duas tabelas em base de dados SQL, neste sistema:

alarm_ID (PK)	...	alarm_ID	...
1	...	1	...
2	...	1	...
3	...	2	...

Tabela 4.2: Tabela de Alarmes, e tabela de Chamadas

Organização da Base de Dados

Por forma a tornar ao texto se tornar mais explicito, introduz-se assim a estrutura da base de dados e as relações existentes entre os diferentes elementos (figura 4.7, página 32).

Acção do Sistema

Tendo em conta estas duas tabelas, atentamos ao seguinte fluxograma (figura 4.8) que explica o procedimento da máquina de estados finitos que resolve a saída de notificações com a entrada de alarmes.

Conforme é descrito, esta máquina de estados inicia a sua tarefa por tratar o primeiro alarme que não foi resolvido e que não teve nenhuma notificação feita. Quando não existir nenhum alarme novo para tratar com notificações a utilizadores, é feito uma recolha dos alarmes que foram feitas chamadas, mas que não houve reconhecimento por parte do utilizador (isto é, chamada não atendida, mensagem de texto não recebida ou sem informação de reconhecimento, chamada falhada...) destes alarmes, é feito de novo uma tentativa de notificação. E o processo repete-se sucessivamente até que não exista nenhuma alarme por notificar.

É importante referir que após confirmação de notificação, o alarme **não** é tratado a menos que fique seja activo novamente. Contudo, se um alarme pendente na lista de alarmes com notificação falhada for entretanto resolvido, não é enviada notificação.

Todos estes *movimentos* são registados em base de dados, para uma posterior pesquisa e verificação de registos de chamadas, alarmes, notificações e ocorrências.

A figura 4.9 procura clarificar todos estes processos de armazenamentos de informação após esta ser “*tratada*”.

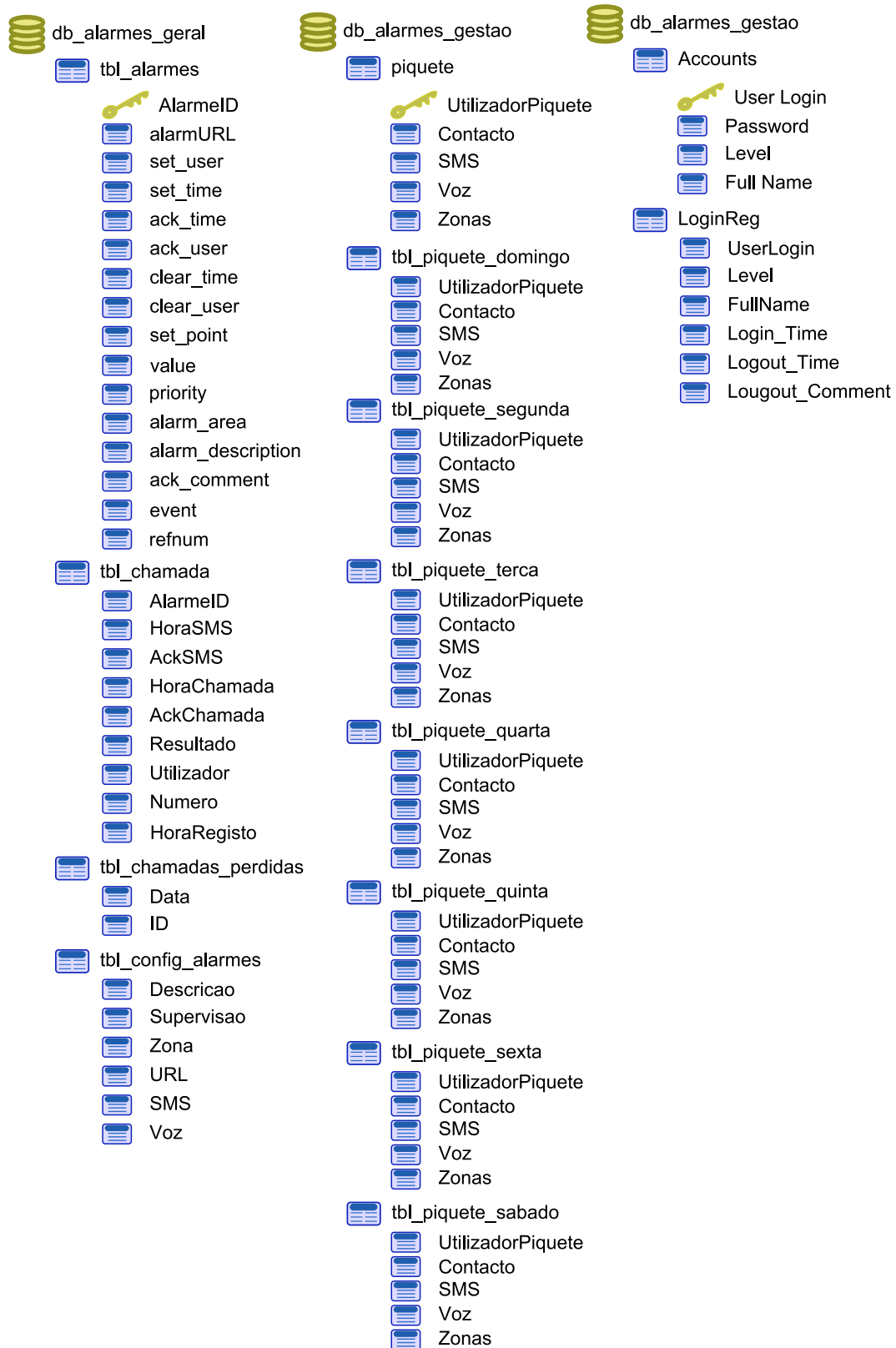


Figura 4.7: Estrutura da base de dados

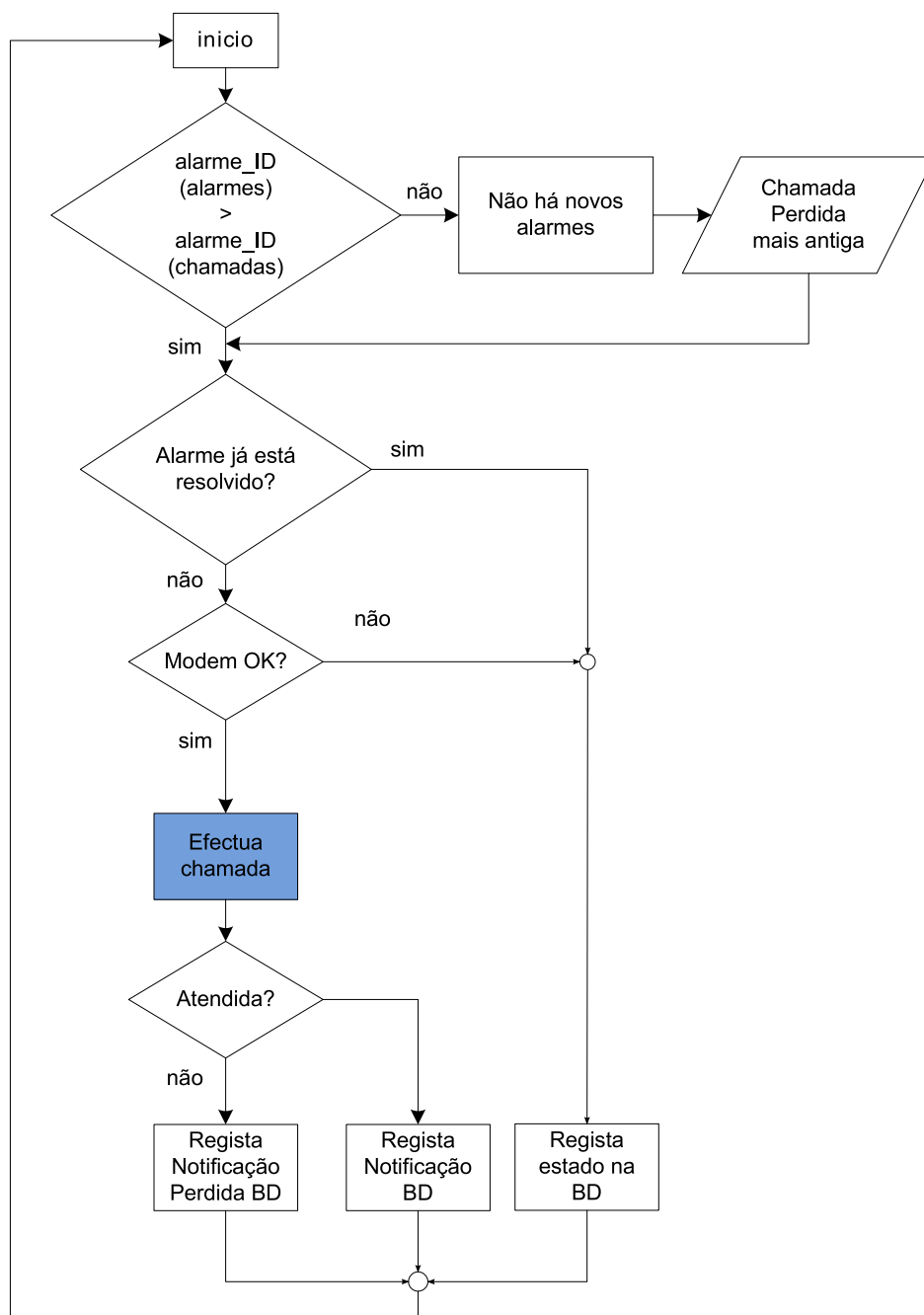


Figura 4.8: Fluxograma com a estrutura da máquina de estados de notificações

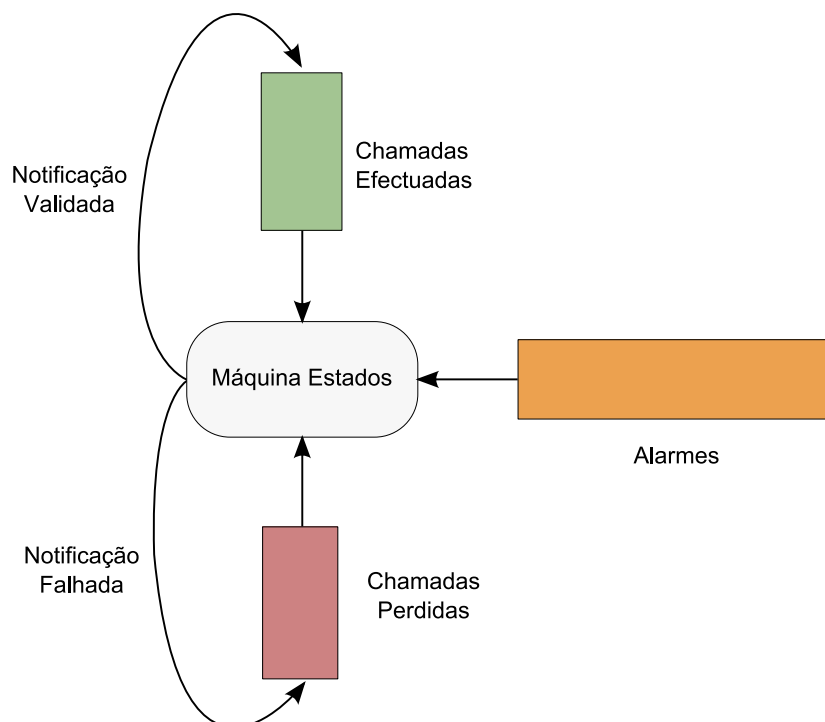


Figura 4.9: *Buffers* e a forma como se interligam com a máquina de estados

4.2.4 Máquina de Estados de Despacho de Notificações

Interligada com a máquina de estados finita anterior (figura 4.8), está outra mais simples, que trata de despachar as notificações que são necessárias enviar para os diferentes utilizadores (esta máquina de estado encontra-se dentro do bloco destacado a azul, na figura 4.8).

Esta máquina de estados pode ser descrita por o fluxograma apresentado na figura 4.10, e é executada enquanto existirem utilizadores registados para a recepção de notificações, por cada alarme.

Vejamos, quando um novo alarme ocorre como é despachado para esta máquina de estados.

Após a identificação do alarme e a escolha do utilizador a notificar, é verificado se o utilizador tem a opção de recepção de chamada, em caso positivo é feita a chamada e registado na base de dados esta accção.

Caso o utilizador não tenha a chamada activa deverá ter a opção de SMS. É escrita uma mensagem escrita e enviada para o utilizador. O sistema aguarda um tempo pré-definido (configurável) por uma chamada deste número. Caso o utilizador não efectue a chamada dentro do tempo definido, o sistema interpreta que a notificação para este utilizador falhou, e repete o procedimento para o próximo utilizador.

Considerando a hipótese que nenhum utilizador reconhece a notificação é feito um registo em base de dados que falharam todas as notificações. Logo que não existir um novo alarme,

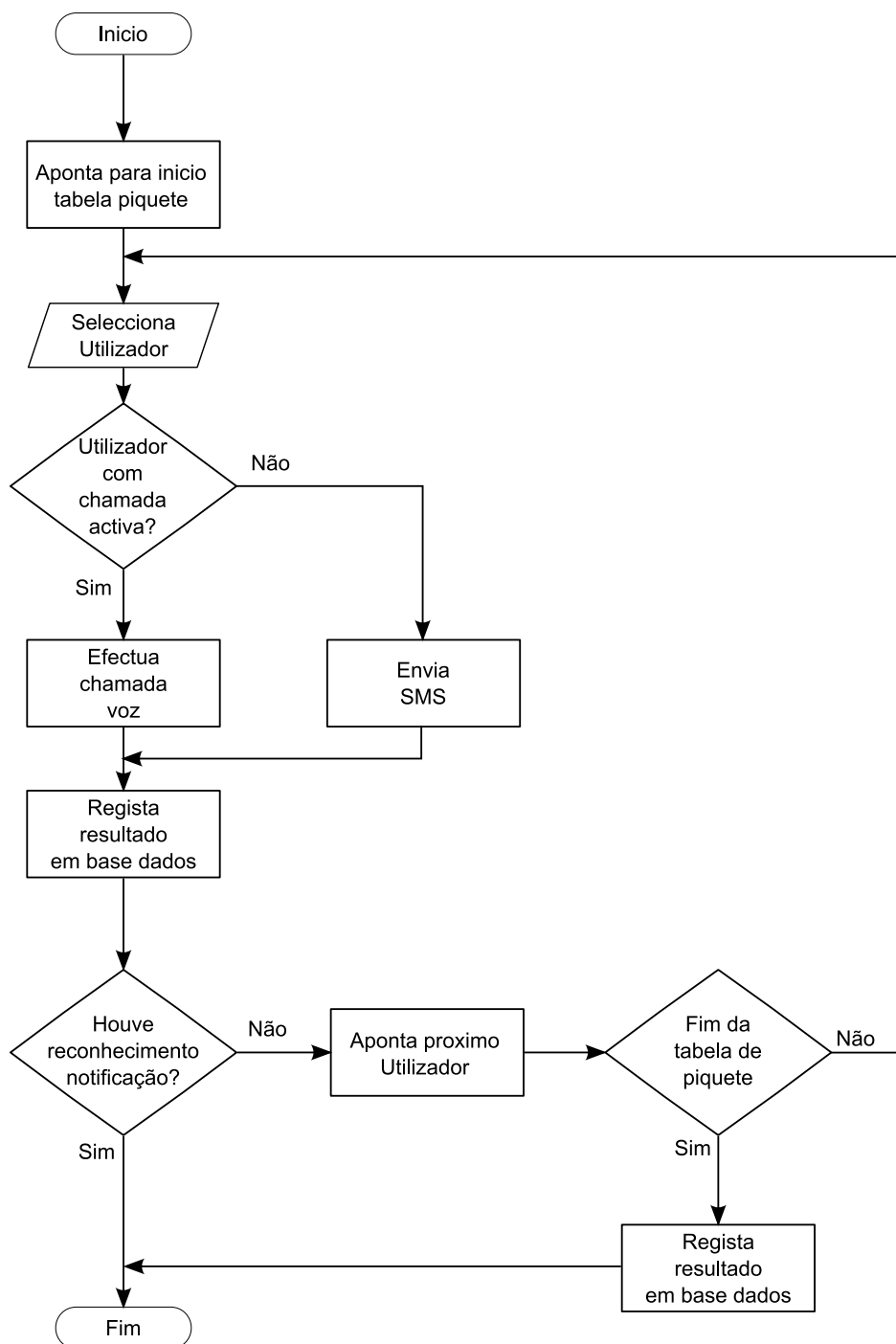


Figura 4.10: Fluxograma com a estrutura da máquina de estados despacho de chamadas/SMS

(e caso este ainda não tenha sido resolvido), os utilizadores serão notificados novamente exactamente com o mesmo processo.

Esta foi a abordagem seguida visto que num ambiente industrial, não é difícil uma chamada telefónica passe despercebida, devido a um grande número de factores externos, tais como ruído, falta de cobertura de rede GSM,...

Todavia, não seria muito interessante nem útil se o sistema apenas tentasse notificar uma vez os utilizadores. A ideia deste trabalho é informar que existe um problema, até que este esteja resolvido, ou até que haja a certeza que o problema foi comunicado correctamente.

Uma vez que as listas de alarmes a serem tratados pelo STI com notificações são configuráveis, apenas os alarmes que requerem uma rápida intervenção são justificáveis de serem tratados por este sistema, não existindo o perigo de o sistema notificar todos os utilizadores por alarmes pouco significativos.

Detalhes das notificações

SMS É enviada uma mensagem escrita com a descrição do alarmes e a zona que é afectada:

1 ALARME! QUADRO ROBOT, FALHA ALIMENTAÇÃO!

Chamada de Voz Requer que o módulo esteja ligado à placa de audio do computador, para este tocar um som, que identifique o alarme. É assim necessário que exista uma personalização dos tipos de alarmes e da mensagem de voz a transmitir. A mensagem de voz é repetida por duas vezes e a chamada é desligada automaticamente após o fim da mensagem.

Destes dois, e pelo conhecimento empírico adquirido ao longo do desenvolvimento do projecto, o mais popular é a mensagem escrita.

- Mais barato
- Facilidade de assimilação da notificação
- Permite que o registo seja guardado para futuras referências
- Comunicação totalmente independente do ambiente sonoro exterior

4.3 Gestão de utilizadores

Esta secção, revela-se bastante importante neste projecto uma vez que mistura conceitos de utilizador, relativamente comuns do dia-a-dia, e um pouco mais específicos como a personalização do software *à la carte*.

4.3.1 Gestão de acessos

A aplicação de Telemanutenção Industrial foi dotado de um sistema de gestão de acessos, de forma a impedir o uso indevido por pessoal de conduta menos ética.

Assim existem sete níveis de utilização, denominados respectivamente:

Sem Utilizador - Apenas pode visualizar os alarmes activos.

Operador - Apenas pode visualizar os alarmes activos

Chefe Turno - Apenas pode visualizar os alarmes activos

Electricista - Apenas pode visualizar os alarmes activos

Chefe Electricista - Pode definir utilizadores e verificar para quem foram enviadas as chamadas.

Chefe Manutenção - Tem as funções do nível anterior, mais as de efectuar pesquisas, configurar tabela de piquete e tabela de alarmes a enviar.

Administrador - Todas as funções anteriores, com a possibilidade de efectuar as configurações do módulo.

Pois até aqui nada de inovador, no entanto, cada uma destes utilizadores, está disponível para fazer parte da **tabela de piquete**.

4.3.2 A tabela de piquete

A tabela de piquete, é como o próprio nome indica uma tabela onde estão agendados e atribuídos os utilizadores que estarão responsáveis por responder com prontidão aos eventuais alarmes que surjam na entidade fabril a fim de otimizar ao máximo o desempenho da fábrica.

Esta tabela estará directamente ligada à tabela de utilizadores de sistema.

A título de curiosidade, é frequente o responsável da secção de manutenção ser o último utilizador registado na tabela de piquete, a fim de garantir o correcto funcionamento da fábrica, e avaliar o desempenho da equipa.

Para o registo de um utilizador genérico na tabela de piquete, existe a possibilidade de filtrar o utilizador por zona da fábrica, a fim de garantir eficiência e rapidez de resolução de problemas.

4.4 Telemantenção

Para este módulo de telemantenção, que se mostra importante na resolução de problemas, foi criado através do LabVIEW um WebServer, que publica o estado actual da aplicação.

Com o intuito de criar uma resposta para este problema simples surgiu a ideia de publicar o serviço com uma página da web, de modo a poder ser visualizado em qualquer parte do mundo sem necessidades de software especiais, tem ainda a vantagem de ser multi-plataforma.

A forma de aceder ao interface e visualizar o estado de todos os alarmes da fábrica, é simplesmente abrir um browser e digitar o endereço de IP em que se encontra o WebServer com a aplicação.

Capítulo 5

Apresentação de resultados

De uma forma simples, todo o trabalho realizado no âmbito deste projecto, representa o resultado obtido, isto é, o resultado deste trabalho é o próprio sistema de Telemantenção Industrial.

Assim este capítulo procura apresentar as aplicações desenvolvidas do STI, procurando mostrar as suas potencialidades.

Esta secção é particularmente rica em *print-screens* do trabalho, de forma a melhor auxiliar a compreensão e funcionamento deste.

5.1 Apresentação dos diferentes módulos

São aqui apresentados os módulos que foram desenvolvidos e aperfeiçoados durante todo o projecto.

Não se procurou ser muito objectivo relativamente ao interface gráfico uma vez que não seria a primeira preocupação do trabalho. Contudo procurou-se adaptar o trabalho as necessidades dos utilizadores e orientar toda a infra-estrutura de modo a que o sistema funcione, mas que seja também útil, e *usável*. Um exemplo prático disso mesmo são os constantes actualizações e aperfeiçoamentos que sofreu desde a ideia original.

Serão apresentadas assim todas as funcionalidades do produto final resultante deste trabalho.

5.2 O Interface

A aplicação é inicializada com dois écrans de arranque do sistema. Estes dois écrans devem-se à inicialização de variáveis na rede e de inicialização da estrutura da base de dados.

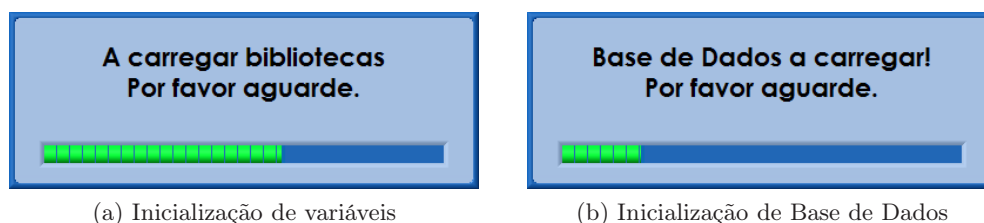
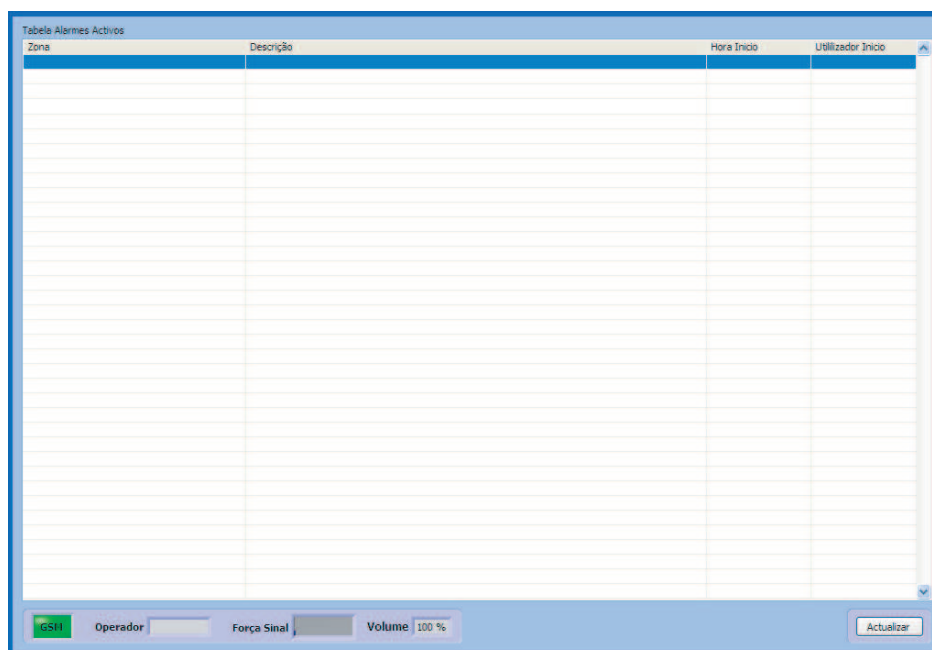


Figura 5.1: Inicialização da aplicação

5.2.1 Tratamento Notificações

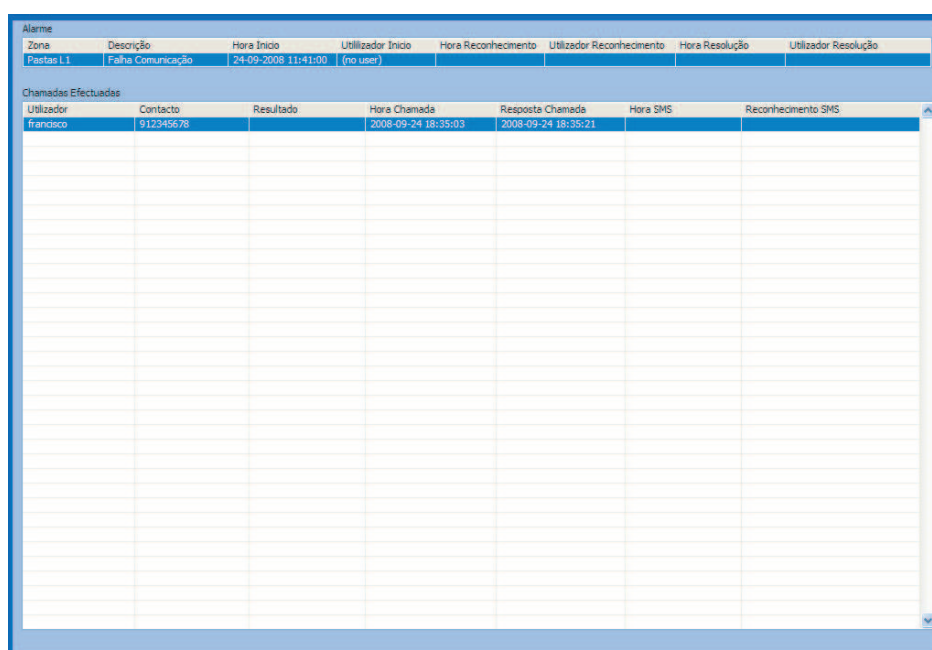
A aplicação começa assim automaticamente, a adquirir alarmes de todos os clientes e a enviar as notificações respectivas. Toda a informação relativa aos alarmes que estão activos em todos os clientes, está disponível numa tabela (figura 5.2) que é actualizada á medida que existem ou são resolvidos os alarmes da fábrica.

Existe a possibilidade de saber quais foram as chamadas (figura 5.3) que foram efectuadas para cada alarme. Para ter acesso a esta informação é necessário *clicar* duas vezes sobre o alarme, e é mostrada a informação relativa as acções que foram tomadas para processar essa notificação.



Zona	Descrição	Hora Inicio	Utilizador Inicio
------	-----------	-------------	-------------------

Figura 5.2: Tabela com alarmes activos



Zona	Descrição	Hora Inicio	Utilizador Inicio	Hora Reconhecimento	Utilizador Reconhecimento	Hora Resolução	Utilizador Resolução
Pastas L1	Falha Comunicação	24-09-2008 11:41:00	(no user)				

Utilizador	Contacto	Resultado	Hora Chamada	Resposta Chamada	Hora SMS	Reconhecimento SMS
francisco	912345678		2008-09-24 18:35:03	2008-09-24 18:35:21		

Figura 5.3: Tabela com as notificações efectuadas para o alarme seleccionado

5.2.2 Módulo de Pesquisa

No meio industrial o *data-logging* é muito importante. Este módulo de pesquisa surge na resposta a essa necessidade (figura 5.4).

Assim com este módulo é possível verificar todos os alarmes que foram registrados pelo sistema. É possível também de forma semelhante ao que foi demonstrado anteriormente, a possibilidade de verificar qual o utilizador notificado, por que meio (chamada/SMS) e qual o resultado da notificação, para um alarme em específico.

Todavia o tratamento de dados de uma pequeno espaço de tempo, é importante e relativamente fácil de processar. Contudo, imaginemos o cenário de uma ritmo de 1 *alarme/segundo* durante apenas um dia (24 *horas*), era necessário pesquisar cerca de 86400 registos, o que se revelaria um pouco difícil. Assim o Sistema de Telemanutenção Industrial oferece ao utilizador filtros de pesquisa, de forma a optimizá-la e a torná-la mais eficiente.

São oferecidos filtros:

- Por data;
- Utilizador;
- Supervisão (cliente);
- Zona;

[illegible]

Figura 5.4: Tabela de histórico de alarmes da fábrica

5.2.3 Filtros de Alarmes

Este foi mais um dos módulos que foi implementado, tendo em consideração algum do conhecimento adquirido no terreno junto do utilizador final do sistema.

A ideia é basicamente aplicar uma filtro aos alarmes que são recebidos pelos sistema, para que não sejam notificados todos os alarmes que surgem.

Assim só é enviada notificação para o utilizador dos alarmes que tiverem sido previamente configurados (figura 5.5).

Descrição	Supervisão	Zona	SMS	Voz
Falta de chama queimador 1	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Temperatura elevada zona queimador 1	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Temperatura baixa zona queimador 1	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Humidade elevada zona queimador 1	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Humidade baixa zona queimador 1	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Controlo manual da temperatura do queimador 1	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Controlo automatico da temperatura do queimador 1	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Controlo manual da humidade do queimador 1	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Controlo automatico da humidade do queimador 1	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Rearme queimador 1	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda temperatura 3	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda temperatura 5	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda temperatura 6	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda humidade 3	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda humidade 5	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda humidade 6	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Falha de ligação	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Falta de chama queimador 2	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Temperatura elevada zona queimador 2	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Temperatura baixa zona queimador 2	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Humidade elevada zona queimador 2	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Humidade baixa zona queimador 2	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Controlo manual da temperatura do queimador 2	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Controlo automatico da temperatura do queimador 2	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Controlo manual da humidade do queimador 2	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Controlo automatico da humidade do queimador 2	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Rearme queimador 2	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda temperatura 7	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda temperatura 8	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda temperatura 9	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda humidade 7	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda humidade 8	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda humidade 9	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>

Figura 5.5: Tabela de configuração de Alarmes

Para este módulo é necessário que haja um levantamento no terreno de todos os alarmes de forma a poder configurar a tabela de alarmes de configuração.

Só serão alvo de notificação os alarmes que forem seleccionados por um utilizador com permissões de administrador de sistema. Todos os outros alarmes serão ignorados pelo bloco de notificações.

5.2.4 Configuração da porta série

A fim de aumentar a portabilidade deste sistema e a torná-lo mais versátil, foi incluído um écran de configuração dos parâmetros de comunicação com o módulo GSM, de forma a garantir que não existem erros no envio das notificações.

São configuráveis:

- a porta de comunicação



Figura 5.6: Tabela de configuração do Modem

- o volume da chamada
- o tempo de espera do sistema para receber confirmação de notificação (caso SMS)

5.3 Os utilizadores

5.3.1 Novo utilizador

Como foi explicado em cima neste texto é conveniente que sejam *barrados* vários tipos de funções a certo tipo de utilizadores.

Apenas para o nível mais alto de utilizador, está disponível a criação de novo utilizador.

Para cada utilizador é imperativo definir os seguintes parâmetros:

- Nome de utilizador;
- Palavra Passe;
- Tipo de utilizador;
- Numero de telefone de contacto;
- Tipo de notificação¹;
- Zona de Acção.

Desta forma é possível obter todas as informações necessárias para que um utilizador possa ser notificado devidamente com o Sistema de Telemantenção Industrial.

5.3.2 Tabela de Piquete

A tabela de piquete é uma funcionalidade que foi adicionada ao projecto, de forma a poder programar o envio de notificações para os utilizadores.

¹É necessário escolher pelo menos um tipo de notificação

- Nome Utilizador
- Contacto
- Notificação SMS (activa/desactiva)
- Notificação Chamada (activa/desactiva)
- Zona 1 (activa/desactiva)
- Zona 2 (activa/desactiva)

Existe ainda a possibilidade de programar as notificações por dias da semana.

5.3.3 Histórico de Utilizações

Uma vez que se trata de uma sistema bastante importante, que visa melhorar o tempo de resposta das equipas de manutenção, é conveniente que exista algo que controle o acesso ao sistema, como já foi abordado. Contudo, a este bloco de acesso ao sistema, foi adicionado um que regista todos os logins que foram efectuados no STI.

Desta feita é possível controlar o quem interveio na aplicação. Todos os *logouts* são comentados permitindo saber se a aplicação foi forçada a ser terminada.

[illegible]

Figura 5.8: Tabela de Pesquisa de Utilizadores

5.4 Limitações do Sistema

Como é inevitável, todo o projecto está sujeito a limitações impostas por os mais variados contextos. Apesar de este ponto ser uma preocupação constante em todo o desenvolvimento deste sistema, surgem sempre factores externos que não serão possíveis de ponderar controlar.

A principal limitação deste trabalho será a dependência do meio de comunicação. Se a infra-estrutura de comunicação não estiver disponível, não será possível obter os dados dos clientes. No entanto esta limitação será considerada uma limitação externa uma vez que não será possível contornar este facto, e contorná-lo desviava o objectivo principal deste trabalho.

É importante referir que a perda de ligação com um dos clientes não afecta a comunicação com qualquer outro dos clientes.

Contudo, o facto de não existir meio de comunicação, não é um factor que deverá condicionar o normal funcionamento que da aplicação cliente, quer da aplicação servidora. Todos os alarmes que ocorrerem em um cliente, são registados num dispositivo de armazenamento de massa, que permite ficar com um histórico dos alarmes da máquina, que estará sempre disponível para consulta posterior.

Assim que a comunicação Cliente → Servidor for restabelecida, o STI, irá despachar estes novos alarmes, caso ainda não tenham sido alvo de notificação.

5.5 Sistema Telemantenção Industrial

Nesta secção são apresentadas várias imagens do sistema computacional STI que foi implementado na Roca em Leiria

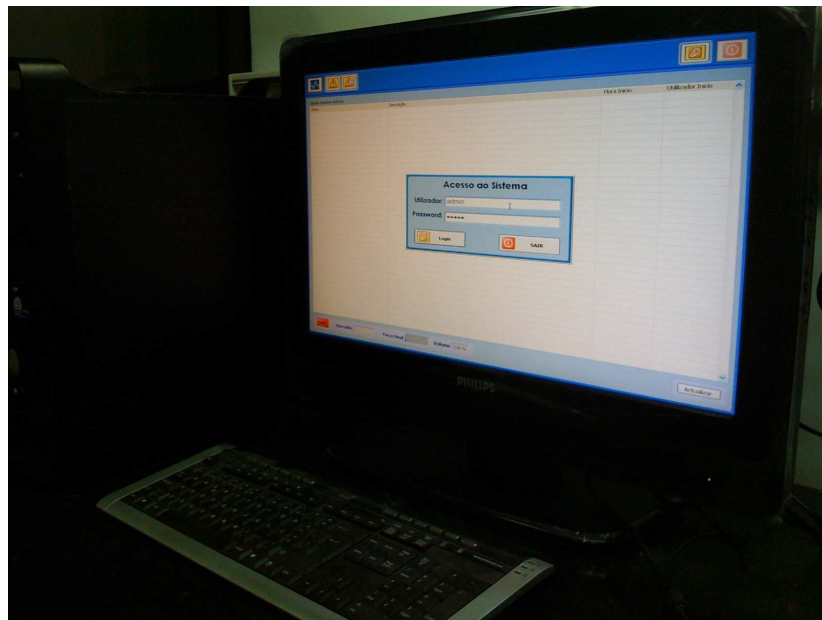


Figura 5.9: Login no sistema

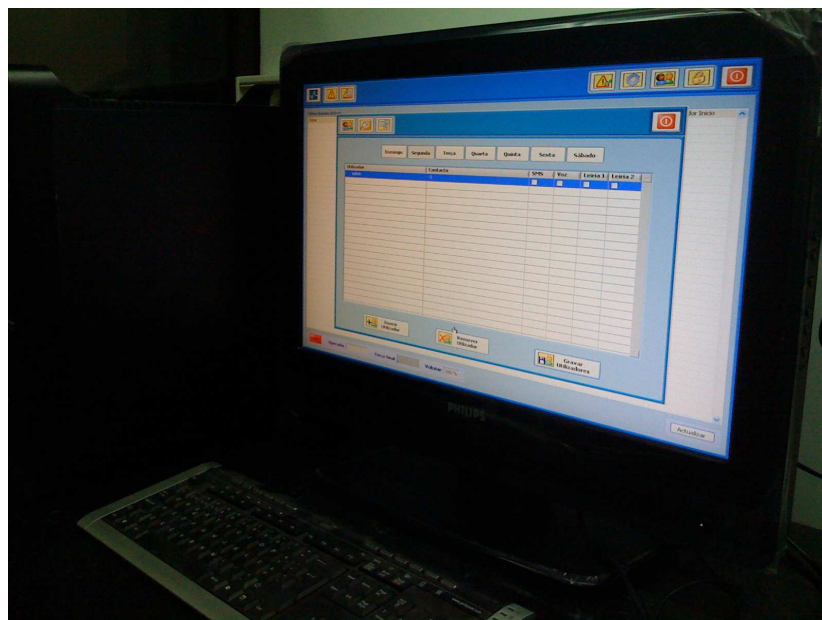


Figura 5.10: Configuração da tabela de Piquete

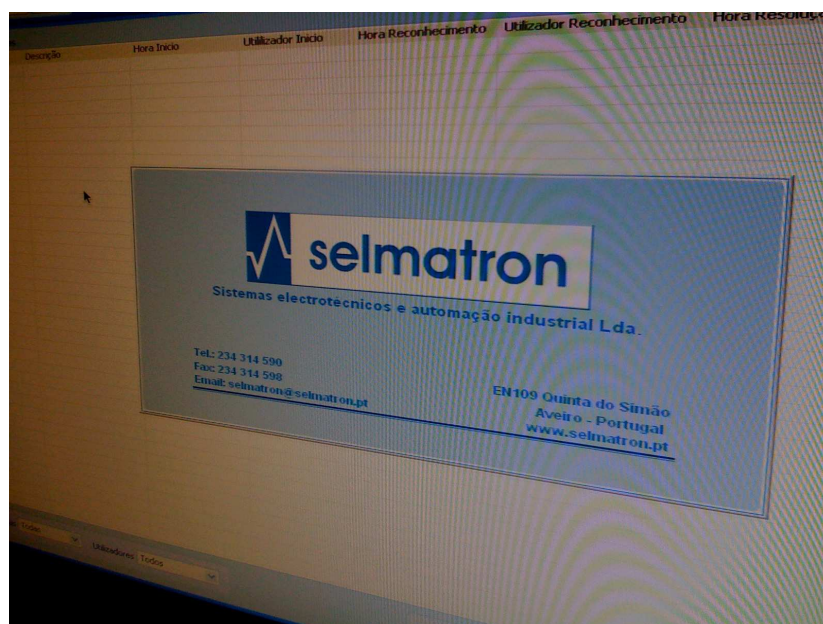
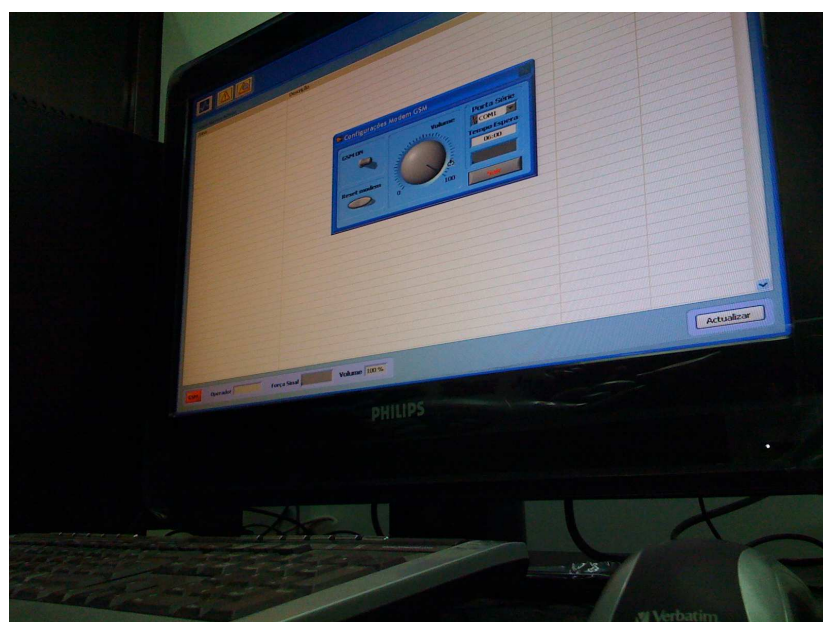


Figura 5.11: Informações do sistema

Figura 5.12: Configuração do *modem* GSM

Capítulo 6

Conclusões

Este capítulo, trata de apresentar as conclusões relativas ao desenvolvimento de todo o projecto.

Engloba também uma secção relativa a a uma proposta do autor relativamente a possíveis trabalhos futuros.

6.1 Discussão geral dos resultados

O trabalho desenvolvido neste projecto teve como finalidade o desenvolvimento de um sistema de telemanutenção que permitisse ter um registo do histórico de alarmes de cada uma das sectores de uma fábrica, e para além disso permitisse avisar a secção de manutenção sobre anomalias nesses sectores.

Através deste projecto verificou-se que é possível desenvolver um sistema simples, seguro e barato que se adapte as necessidades de cada entidade fabril.

Este sistema justifica a sua viabilidade de implementação através:

Preço O custo no meio empresarial é sempre muito importante. Tendo em conta este facto, o sistema de telemanutenção desenvolvido o seu custo de implementação seria basicamente na aquisição de do sistema composto por uma unidade computacional e módulo GSM. É obvio que este ponto representa sempre uma despesa para a empresa que decide implementar este sistema, contudo será rapidamente compensado uma vez que serão reduzidos *tempos de máquina parada*.

Simplicidade Foi um ponto de preocupação constante no desenvolvimento do projecto, o facto de elaborar um sistema simples, a vários níveis, quer a nível computacional, quer a nível de operação, uma vez que a sua função é permitir a fácil resolução de problemas, toda a complexidade deveria ser *escondida* por um interface simples e prático.

Desta forma este sistema, mostra-se uma boa opção para empresas com grandes unidades industriais, uma vez que além de alertar para o problema, localiza-o, optimizando assim o tempo entre o acontecimento da anomalia e o inicio da intervenção.

6.2 Trabalhos futuros

Um factor a ser explorado, e apoiado neste projecto será a possibilidade de elaboração de um sistema de de telemanutenção semelhante, a mais *baixo-nível*, isto é, construir sistemas mais pequenos e portáteis, com funções um pouco mais simples que permitam serem adaptados a sistemas com necessidades semelhantes mas que não englobem na sua composição uma unidade computacional, que é um factor essencial na aplicação deste projecto.

Outra proposta de trabalho seria a integração com um projecto para telemanutenção, isto é, uma unidade remota permitindo supervisão de variáveis, apenas com o módulo GSM, permitindo assim o desenvolvimento de uma plataforma muito simples, com elevadas potencialidades.

Anexos

Capítulo 7

Manual do Utilizador

Este capítulo destina-se a explicar de uma forma simples como utilizar a aplicação desenvolvida no âmbito deste projecto.

7.1 Início

A aplicação começa inicializar as bibliotecas de comunicação, de seguida são carregadas as bases de dados necessárias para gravar os vários registos do sistema e por fim é iniciado o sistema de gestão de acessos.

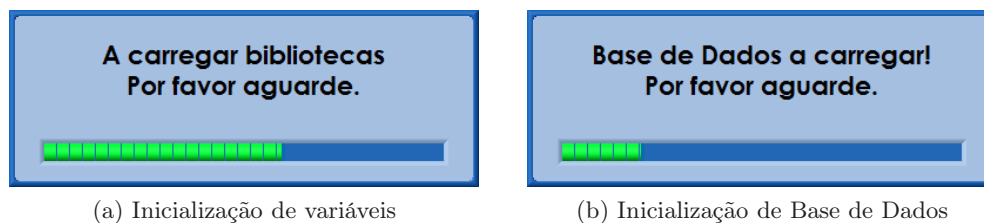


Figura 7.1: Inicialização da aplicação

7.2 Menu de Sistema

Na figura 7.2, podem-se encontrar os botões de acesso as diferentes funções do *software*, com a respectiva descrição.

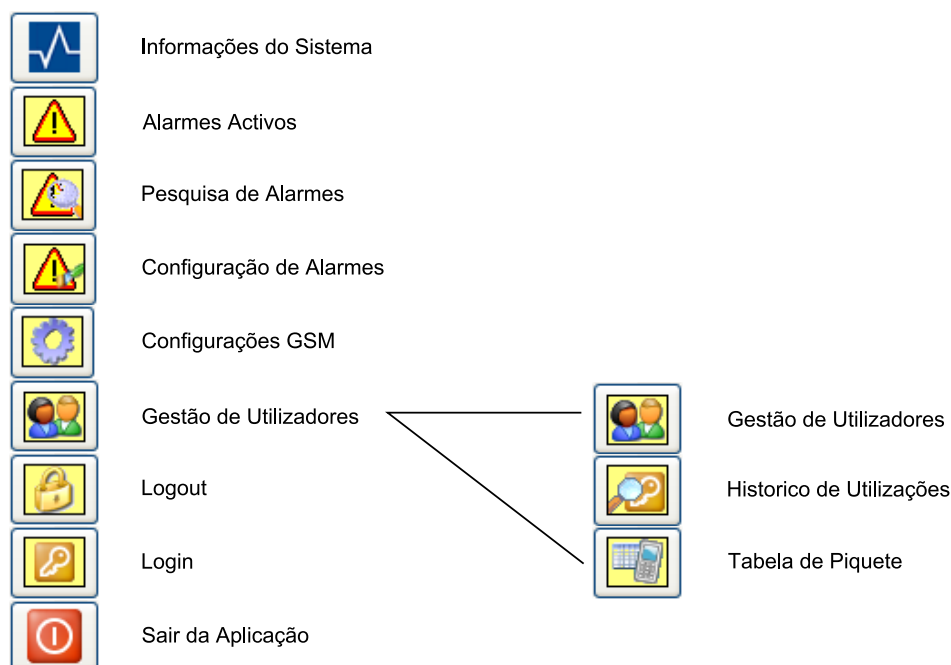


Figura 7.2: Menu do Sistema

7.3 Écran Principal

Neste écran é visível uma tabela que tem em tempo real toda a informação dos alarmes activos.

[illegible]

Figura 7.3: Tabela de histórico de alarmes

Clicando duas vezes sobre um determinado alarme, a aplicação passa para outro écran que permite ver quais as chamadas que foram efectuadas para esses mesmo alarme (no caso de ter sido efectuada alguma.)

7.4 Pesquisa de Alarmes

Neste écran do software é possível efectuar uma pesquisa de todo o histórico dos alarmes registados, sendo possível filtrar a pesquisa por:

- Zona
- Utilizador
- Por janela temporal

Desta forma é permitido ao utilizador do software que optimize a sua pesquisa.

É também disponibilizada a funcionalidade semelhante a do écran de alarmes onde o utilizador pode verificar que chamadas foram efectuadas para o alarme que seleccionar com duplo clique.

Todos Alarmes				
Descrição	Supervisão	Zona	SMS	Voz
Falta de chama queimador 1	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Temperatura elevada zona queimador 1	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Temperatura baixa zona queimador 1	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Humidade elevada zona queimador 1	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Humidade baixa zona queimador 1	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Controlo manual da temperatura do queimador 1	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Controlo automatico da temperatura do queimador 1	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Controlo manual da humidade do queimador 1	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Controlo automatico da humidade do queimador 1	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Rearme queimador 1	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda temperatura 3	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda temperatura 5	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda temperatura 6	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda humidade 3	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda humidade 5	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda humidade 6	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Falha de ligação	ClimaMedelena	Queimador 1	<input type="checkbox"/>	<input type="checkbox"/>
Falta de chama queimador 2	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Temperatura elevada zona queimador 2	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Temperatura baixa zona queimador 2	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Humidade elevada zona queimador 2	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Humidade baixa zona queimador 2	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Controlo manual da temperatura do queimador 2	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Controlo automatico da temperatura do queimador 2	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Controlo manual da humidade do queimador 2	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Controlo automatico da humidade do queimador 2	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Rearme queimador 2	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda temperatura 7	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda temperatura 8	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda temperatura 9	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda humidade 7	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda humidade 8	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>
Falha sonda humidade 9	ClimaMedelena	Queimador 2	<input type="checkbox"/>	<input type="checkbox"/>

Figura 7.5: Configuração de Alarmes

7.7 Configurações do Modem GSM

Écran apenas disponível para o administrador da aplicação

Neste *pop-up*, são dadas a seguintes possibilidades de configuração ao utilizador:



Figura 7.6: Tabela de configuração do Modem GSM

- Ligar/Desligar GSM
- Reset ao Modem
- Volume da Chamada

- Porta de Comunicação Série
- Tempo de espera por Reconhecimento da SMS
- Verificar a potência do sinal de rede

7.8 Gestão de Acessos

Écran apenas disponível para o administrador de sistema

Neste écran são visíveis os utilizadores registados, a respectiva identificação e os níveis. É possível, criar novo, editar, ou apagar utilizador

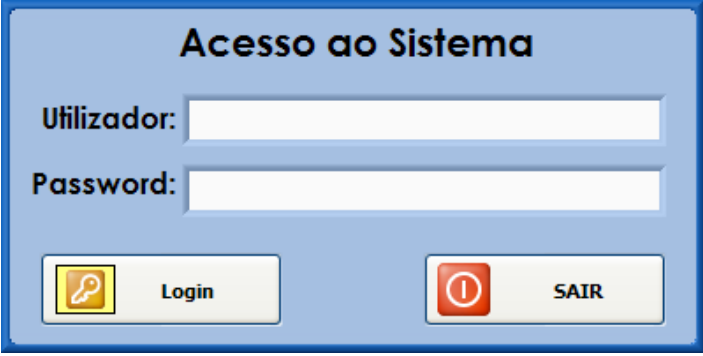
[illegible]

Figura 7.7: Tabela de Gestão de Logins

7.8.1 Novo Utilizador

Neste écran pode-se criar um novo utilizador, para isso basta o utilizador introduzir os seguintes dados:

- Utilizador (Login)
- Password
- Nível de Utilizador



Acesso ao Sistema

Utilizador:

Password:



 Login  SAIR

Figura 7.8: Logins



Novo Utilizador

Utilizador:

Password:

Nivel: ▼

Nome:

Contacto: Zona 1 ☐

SMS ☐ Voz ☐ Zona 2 ☐



 Novo Utilizador  SAIR

Figura 7.9: Novo Utilizador

- Nome do Utilizador
- Contacto
 - SMS
 - Voz
 - Fabrica 1
 - Fabrica 2

7.8.2 Edição de Utilizador



Figura 7.10: Edição de Utilizador

De forma análoga é possível editar um utilizador já existente, bastando para tal o utilizador seleccionar no écran de gestão de acessos o utilizador que pretende editar, seleccionar o botão editar utilizador e o sistema mostra o écran de edição onde aparecerão os valores anteriormente definidos, bastando ao utilizador alterar esses mesmos valores. Por fim carregar no botão guardar utilizador.

7.8.3 Actividade dos Utilizadores

Este écran pode ser entendido também com uma pesquisa de logins.

Assim o utilizador poderá procurar numa determinada data quais os utilizadores que fizeram login, logout e qual o comentário de logout, isto é, se foi feito logout pelo utilizador ou se a aplicação foi encerrada incorrectamente.

Esta pesquisa permitir filtros por:

- Utilizador
- Nível
- Data

7.8.4 Tabela de Piquete

Esta tabela mostra e permite a programação dos utilizadores que serão notificados quando o sistema está a funcionar.

É possível, inserir, remover, e gravar utilizadores na tabela.

[illegible]

Figura 7.11: Tabela de Piquete

É também possível operar a tabela com o sistema em cursos sem que exista qualquer tipo de conflito para as notificações.

Por defeito a tabela de piquete é carregada no dia actual.

7.9 Sair da Aplicação

O sistema questiona o utilizador se pretende encerrar a aplicação em caso afirmativo o sistema fecha aplicação, em caso negativo este volta ao écran anterior.

Capítulo 8

Código Fonte

Neste Capitulo será apresentado o código da aplicação.

É importante referir que o código da aplicação é propriedade da Selmatron, Lda não podendo ser distribuido ou reporduzido.

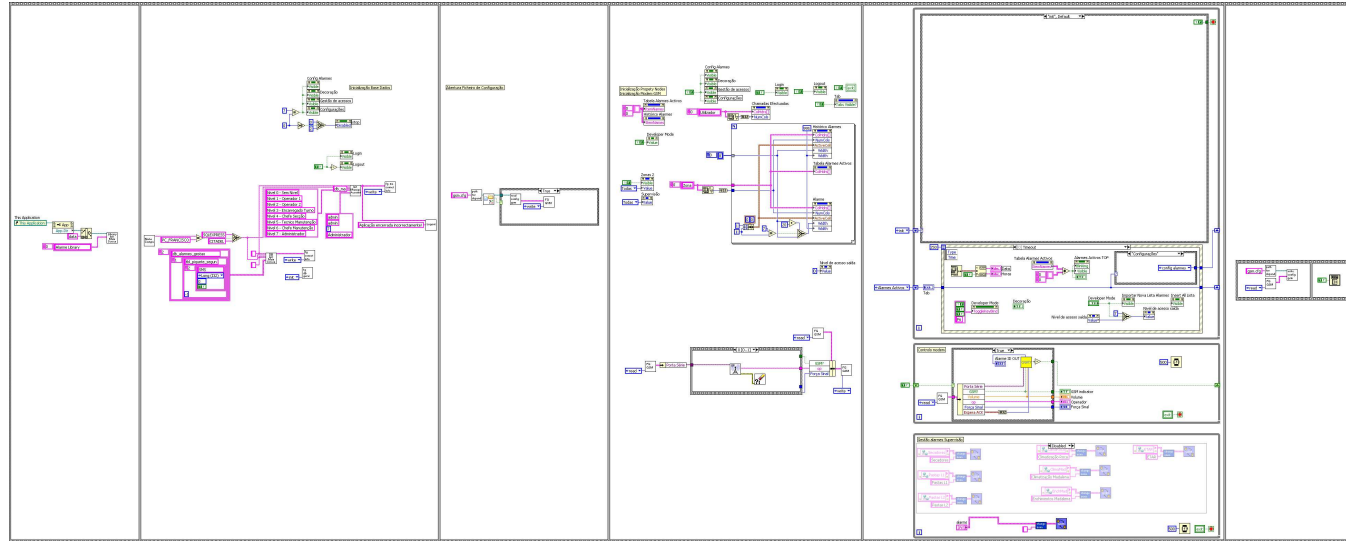


Figura 8.1: main.vi

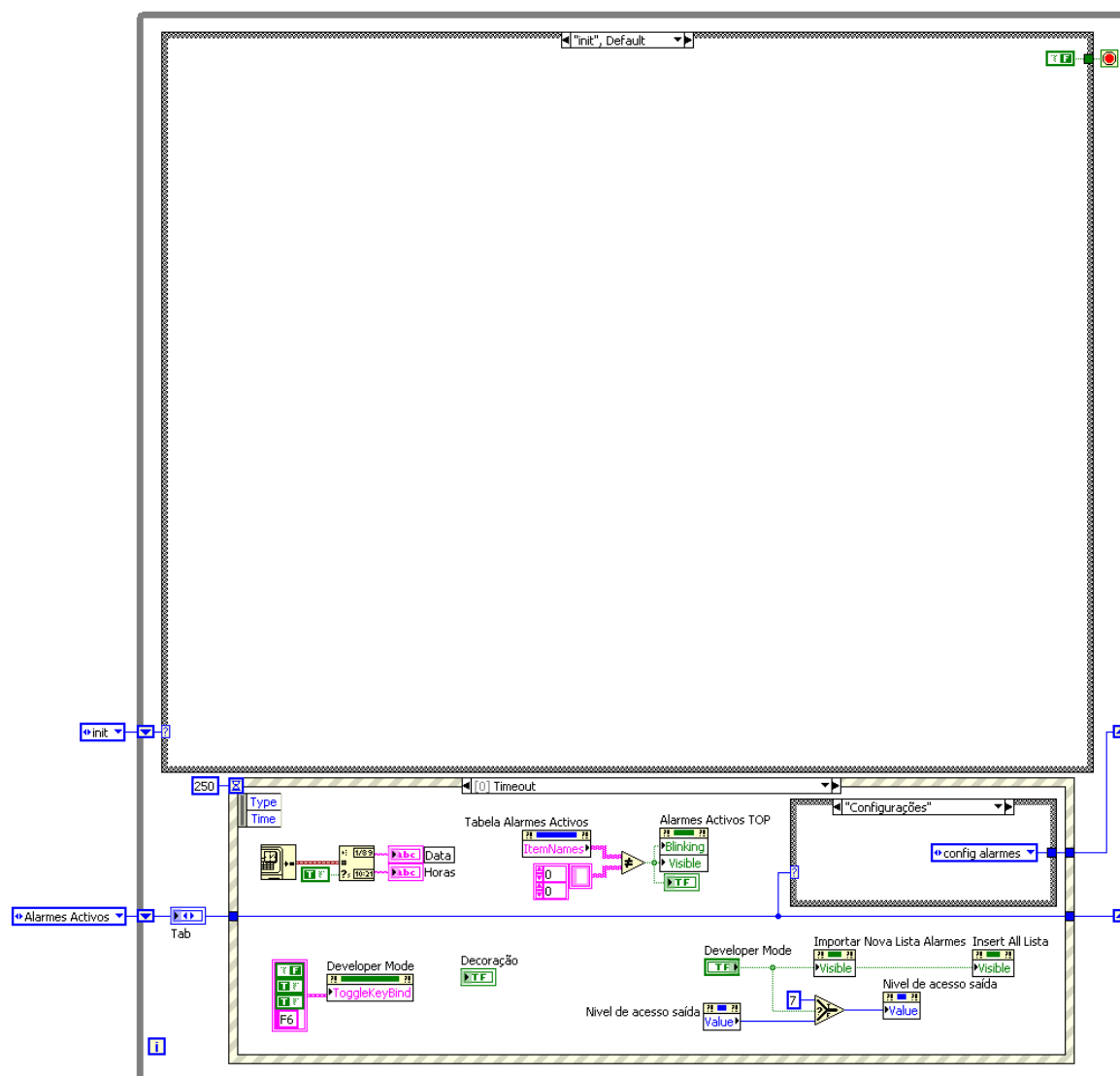


Figura 8.2: Controlo de interface com o utilizador

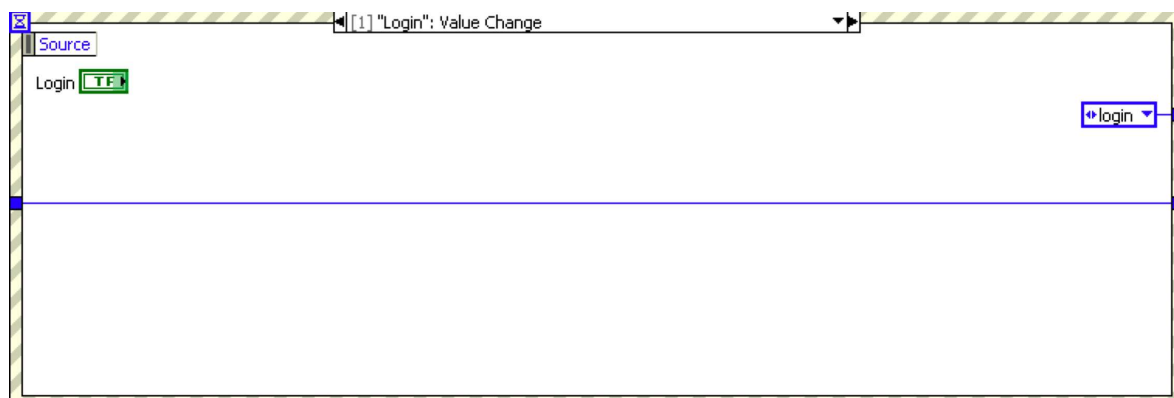


Figura 8.3: Evento de Login

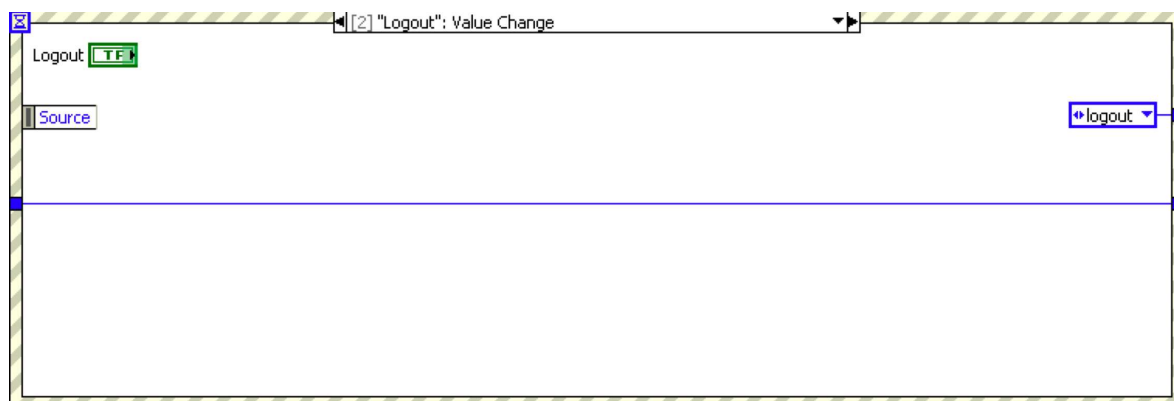


Figura 8.4: Evento de Logout

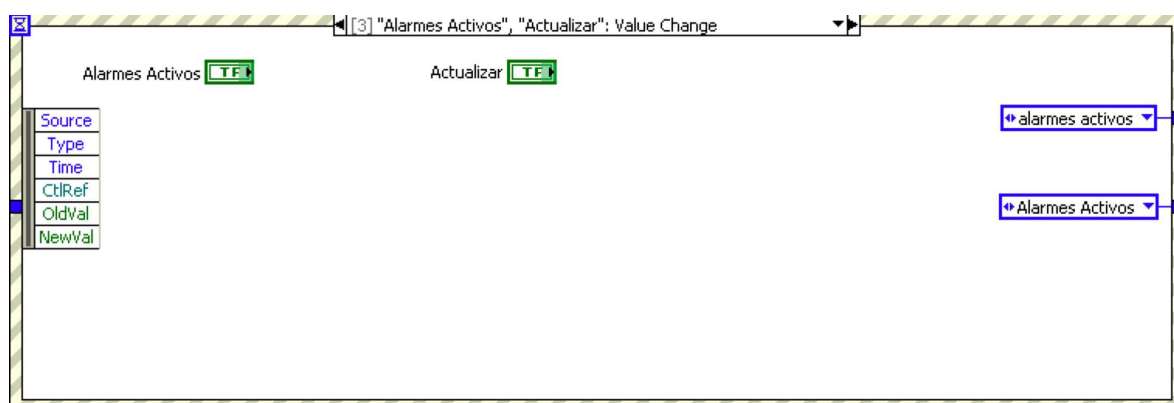


Figura 8.5: Evento que mostra os alarmes activos

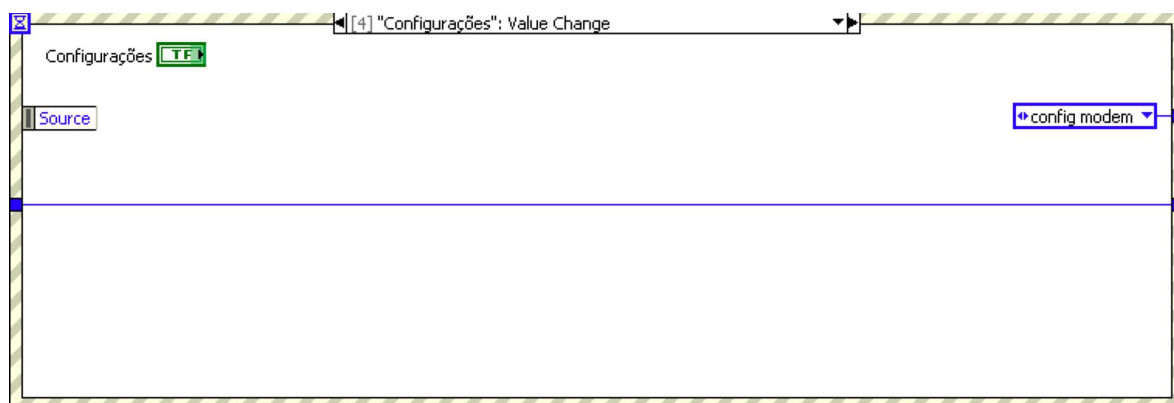


Figura 8.6: Configurações do módulo

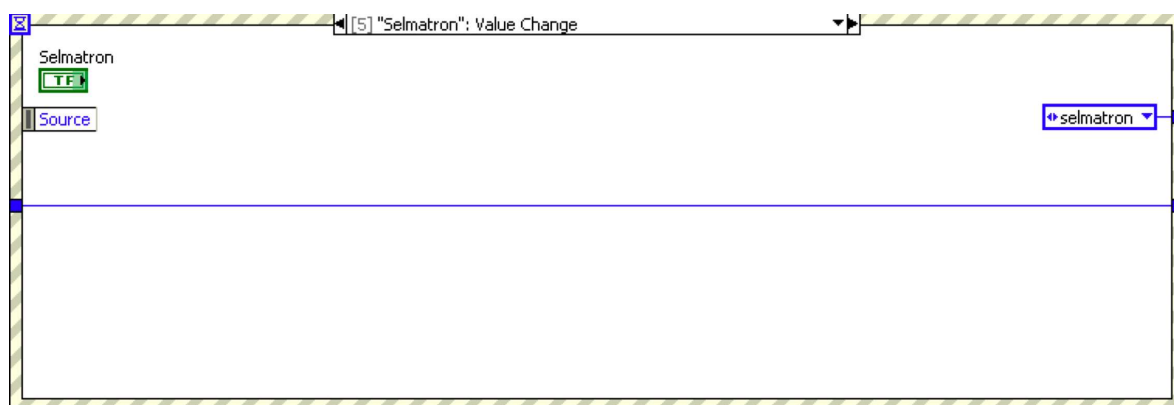


Figura 8.7: Informações da Selmatron

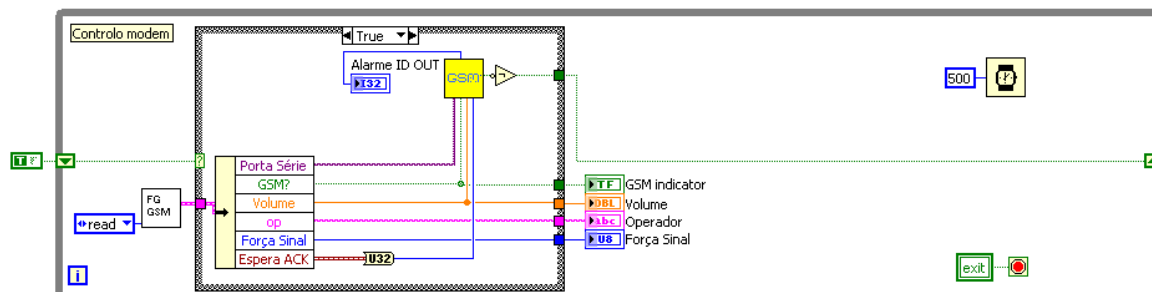


Figura 8.8: Máquina de Estados GSM

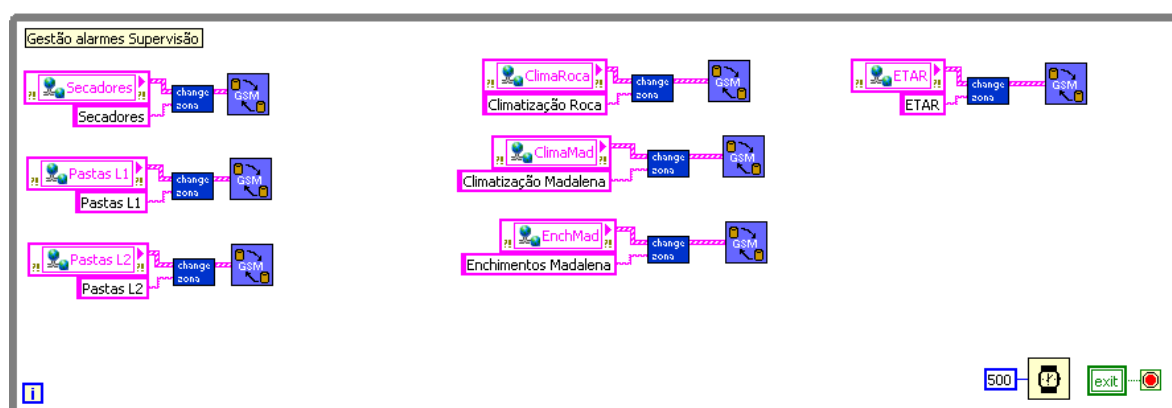


Figura 8.9: Aquisição de dados de cada uma das supervisões da fábrica

Bibliografia

- [1] **National Instruments**, *LabVIEW: User Manual* National Instruments, 2008
- [2] **Rick Bitter, Taqi Mohiuddin, Matt Nawrocki** *LabVIEW, Advanced Programming Techniques, 2nd Edition* CRC Press, 2006
- [3] **Gary W. Johnson, Richard Jennings**, *LabVIEW Graphical Programming* McGraw-Hill Professional, 2006
- [4] **David J. Ritter**, *LabVIEW GUI: Essential Techniques* McGraw-Hill Professional, 2001
- [5] **Leonard Sokoloff**, *Applications in LabVIEW* Pearson/Prentice-Hall, 2003
- [6] **Ying Bai**, *The Windows Serial Port Programming Handbook* CRC Press, 2004
- [7] **Caroline M. Halliday, Hayes Microcomputer Products, Inc.**, *Official Hayes Modem Communications Companion* IDG Books, 1994
- [8] **Kevin E. Kline, Brand Hunt**, *SQL in a Nutshell: A Desktop Quick Reference* O'Reilly, 2004
- [9] **Anthony Molinaro**, *SQL Cookbook* O'Reilly, 2005
- [10] **Alan Beaulieu**, *Learning SQL* O'Reilly, 2005
- [11] **Jonathan Gennick**, *SQL Pocket Guide: SQL Syntax and Usage* O'Reilly, 2006

